

GRADO EN INGENIERÍA MULTIMEDIA



VNIVERSITAT
DE VALÈNCIA

TRABAJO FIN DE GRADO

DONA'M MÓN: APLICACIÓN MÓVIL PARA
VISIBILIZAR LA VIDA Y OBRA DE MUJERES EN
PUNTOS GEOLOCALIZADOS DE VALENCIA.

AUTOR:
IVANA STANISLAVOVA IVANOVA

TUTOR:
MANUEL PÉREZ AIXENDRI



VNIVERSITAT
DE VALÈNCIA [○*] Escola Tècnica Superior
d'Enginyeria ETSE-UV

TRABAJO FIN DE GRADO

DONA'M MÓN: APLICACIÓN MÓVIL PARA
VISIBILIZAR LA VIDA Y OBRA DE MUJERES EN
PUNTOS GEOLOCALIZADOS DE VALENCIA.

AUTOR: IVANA STANISLOVA IVANOVA

TUTOR: MANUEL PÉREZ AIXENDRI

Declaración de autoría:

Yo, Ivana Stanislavova Ivanova, declaro la autoría del Trabajo Fin de Grado titulado “DONA’m MÓN: Aplicación móvil para visibilizar la vida y obra de mujeres en puntos geolocalizados de Valencia.” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.

Valencia, 16 de junio de 2025

Fdo: Ivana Stanislavova Ivanova

Resumen:

Este es el resumen del TFM. Debe ser corto (máximo media página) y cubrir los aspectos principales del TFM.

Abstract:

This is the abstract of the TFM. It must be short and cover the main aspects of the TFM.

Resum:

Aquest és el resum del TFM. Ha de ser curt (màxim mitja pàgina) i cobrir els aspectes principals del TFM.

Agradecimientos:

Índice general

1. Introducción	19
1.1. Introducción	19
1.2. Motivación	19
1.3. Objetivos	20
1.4. Organización de la memoria	21
2. Estado del arte	23
2.1. Historia, evolución e impacto de las aplicaciones móviles	23
2.1.1. Origen y evolución de los dispositivos móviles	23
2.1.2. Ecosistemas y sistemas operativos móviles	24
2.1.3. Tipos de aplicaciones móviles	24
2.1.4. Impacto económico y social	25
2.2. Evolución de la Realidad Aumentada en el Contexto de las Aplicaciones Móviles	26
2.2.1. Fundamentos y primeras ideas (1960–1990)	26
2.2.2. Definición formal y primeros prototipos (1990–2000)	27
2.2.3. Herramientas abiertas y salto a lo portátil (1999–2010)	27
2.2.4. Consolidación y tendencias actuales (2010–2025)	28
2.3. La visibilización de las mujeres en el espacio urbano: una cuestión de justicia histórica	29
2.3.1. Referentes femeninos en la ciudad: biografías y puntos de interés . .	30
2.4. Análisis de aplicaciones similares	32
2.4.1. Pokémon GO	32
2.4.2. Streetmuseum	32
2.4.3. CultuAR	33
2.4.4. Historypin	33
2.5. Tecnologías	34
2.5.1. Plataformas de desarrollo de aplicaciones móviles	34
2.5.2. Geolocalización	36
2.5.3. Realidad Aumentada (RA)	37

2.5.4. Base de Datos	39
2.5.5. Arquitectura de persistencia y gestión de datos	39
3. Requisitos, especificaciones, coste, riesgos, viabilidad	43
3.1. Requisitos	43
3.1.1. Requisitos funcionales	43
3.1.2. Requisitos funcionales	43
3.1.3. Requisitos no funcionales	45
3.2. Especificaciones	46
3.3. Ciclo de vida del software	48
3.4. Planificación temporal	49
3.4.1. Descomposición de tareas por fases	49
3.4.2. Estimación temporal	51
3.4.3. Diagrama de Gantt	52
3.5. Estimación de costes	54
3.5.1. Costes directos de personal	54
3.5.2. Asignación temporal por perfil	55
3.5.3. Costes directos de material	55
3.5.4. Costes indirectos	56
3.5.5. Coste total del proyecto	56
3.6. Viabilidad del proyecto	57
3.6.1. Viabilidad económica	57
3.6.2. Viabilidad legal	57
3.7. Análisis de riesgos	58
3.7.1. Principales riesgos identificados	58
3.7.2. Matriz de riesgos	59
3.7.3. Resumen de evaluación de riesgos	60
3.7.4. Estrategias de mitigación	60
3.7.5. Planes de contingencia	61
4. Análisis	63
5. Diseño	65
6. Implementación y pruebas	67
6.1. Implementación	67
6.2. Pruebas funcionales	67
6.3. Pruebas de rendimiento	67

6.4. Pruebas de usabilidad	67
7. Conclusiones	69
7.1. Revisión de costes	69
7.2. Conclusiones	69
7.3. Trabajo futuro	69
A. Apéndice	71
A.1. Ejemplos del lenguaje de marcado Latex	71
Bibliografía	72

Capítulo 1

Introducción

1.1. Introducción

En la actualidad, la tecnología se ha convertido en una herramienta clave para transformar la forma en la que interactuamos con nuestro entorno, permitiendo nuevas maneras de aprender, explorar y conectarnos con la historia y la cultura. Sin embargo, la construcción de los relatos históricos y culturales ha tendido a otorgar mayor protagonismo a figuras masculinas, dejando en la sombra numerosas contribuciones femeninas de gran relevancia en ámbitos como la ciencia, el arte, la política o la educación.

Este Trabajo de Fin de Grado se centra en el desarrollo de una aplicación móvil interactiva basada en geolocalización y realidad aumentada (RA), que tiene como objetivo visibilizar el legado de mujeres con relevancia histórica que han nacido, vivido, desarrollado parte de su labor o tenido relevancia en Valencia. Mediante esta aplicación, los usuarios podrán desbloquear contenido educativo y multimedia al visitar puntos de interés geolocalizados, conectando los lugares emblemáticos de la ciudad con las historias de estas mujeres, conociendo su vida y obra.

La aplicación lleva por nombre DONA'm MÓN, una frase en valenciano con doble significado en su primera palabra: “dona” puede leerse tanto como sustantivo (“mujer”) o como forma verbal (“dame”). Así que el título leído como “Dame mundo”, esconde un juego de palabras que representa el propósito del proyecto: ofrecer a las mujeres este espacio real y simbólico en el mundo, un lugar en el mapa donde su historia pueda ser descubierta y reconocida.

DONA'm MÓN ofrece una experiencia educativa y lúdica que invita a reflexionar sobre la igualdad de género. Este proyecto pretende ser un puente entre el pasado y el presente, ayudando a los usuarios a redescubrir la ciudad desde esta nueva perspectiva inclusiva.

1.2. Motivación

El desarrollo de este proyecto surge de una doble motivación: por un lado, mi interés personal en el desarrollo de tecnologías con un propósito educativo y social, y por otro, el deseo de poner en valor las aportaciones de las mujeres a lo largo de la historia.

Durante mi recorrido académico como estudiante de Ingeniería Multimedia y en mi máster actual en Tecnologías web, Aplicaciones móviles y Computación en la nube, he desarrollado un fuerte interés por aplicar el conocimiento técnico en proyectos que tengan

un impacto positivo en la sociedad, orientados a transformar la manera en la que accedemos y transmitimos el conocimiento.

Además, como mujer en STEM, considero importante que este proyecto consiga ayudar en la visibilización del papel femenino en todas las áreas del saber y a toda clase de público. Aunque se han producido avances significativos en materia de igualdad, los datos actuales evidencian que la brecha persiste: solo el 33 % de los investigadores a nivel mundial son mujeres, y en campos como la inteligencia artificial esta cifra desciende al 22 % [1]. En el contexto español, apenas el 24 % de las cátedras universitarias están ocupadas por mujeres, según el informe “Científicas en Cifras 2021” del Ministerio de Ciencia e Innovación [2]. Estas cifras reflejan una infrarrepresentación que no solo afecta a la ciencia, sino también a otros ámbitos clave del desarrollo cultural y social.

Valencia, como muchas otras ciudades, alberga una historia rica y compleja en la que también han participado numerosas mujeres valiosas, desde científicas y artistas hasta activistas y educadoras. Muchas de ellas siguen siendo poco conocidas fuera de círculos especializados. Este proyecto representa una oportunidad para darles mayor protagonismo, aprovechando tecnologías como la realidad aumentada y la geolocalización para generar experiencias de aprendizaje activas, accesibles e innovadoras.

Por todo esto, el objetivo de este trabajo no es solo técnico, sino también ético y cultural: contribuir, desde la tecnología, a construir una memoria más justa e inclusiva.

1.3. Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación móvil interactiva que utilice tecnologías de geolocalización, realidad aumentada (RA) y una base de datos centralizada para visibilizar y divulgar las historias de mujeres destacadas en la ciudad de Valencia, promoviendo su reconocimiento histórico y cultural a través de una experiencia inmersiva, educativa y accesible para los usuarios.

Descomponiendo este objetivo principal, se definen los siguientes objetivos específicos:

- **Investigación y selección de mujeres invisibilizadas:**

- Identificar figuras femeninas históricas relacionadas con Valencia que hayan destacado en campos como la ciencia, el arte, la política o los derechos sociales.
- Asociar a cada figura un punto geolocalizado en la ciudad con un vínculo significativo con su historia.

- **Creación y gestión de una base de datos con MySQL:**

- Diseñar y desarrollar una base de datos relacional en MySQL, estructurada para almacenar de forma eficiente información sobre las mujeres seleccionadas, sus ubicaciones geográficas y el contenido multimedia vinculado.
- Implementar un backend utilizando el framework Django, asegurando la integridad, consistencia y seguridad de los datos mediante su ORM, y exponiendo una API RESTful que permita su acceso y manipulación desde la aplicación móvil.

- **Desarrollo de funcionalidades tecnológicas:**

- Implementar un sistema de geolocalización que permita a los usuarios desbloquear contenido interactivo al acercarse a los puntos de interés.
 - Integrar elementos de realidad aumentada desarrollados con A-Frame, que permitan visualizar objetos, imágenes o recreaciones 3D relacionadas con las mujeres seleccionadas al enfocar con la cámara del dispositivo móvil.
 - Implementar un sistema de notificaciones que alerte a los usuarios al aproximarse a un punto de interés relevante, mejorando la interacción con el entorno urbano y fomentando la exploración activa.
- **Diseño de una experiencia de usuario intuitiva y atractiva:**
- Crear una interfaz accesible y adaptada al entorno móvil con React Native y Expo Go, que facilite la navegación y el uso de las funcionalidades principales de la aplicación.
 - Incluir un mapa interactivo con los puntos de interés marcados, accesibles según la ubicación del usuario.
- **Generación de contenido multimedia:**
- Diseñar materiales interactivos y educativos como textos, audios, imágenes, videos y modelos 3D que permitan a los usuarios conocer la vida y las contribuciones de las mujeres seleccionadas.
 - Desarrollar un sistema de recompensas o logros para incentivar la exploración de todos los puntos geolocalizados.
- **Compatibilidad multiplataforma y pruebas de usabilidad:**
- Garantizar que la aplicación sea funcional en dispositivos Android e iOS mediante el uso de herramientas multiplataforma como React Native y Expo Go.
 - Realizar pruebas de usabilidad con un grupo de usuarios para evaluar la experiencia de uso y verificar que se cumplen los objetivos educativos y tecnológicos planteados.

1.4. Organización de la memoria

Esta memoria se estructura en siete capítulos principales, además de un apéndice y una bibliografía final. A continuación se describen brevemente los contenidos de cada uno de ellos:

Capítulo 1 – Introducción: contextualiza el trabajo realizado, presentando la motivación personal y social del proyecto, así como los objetivos generales y específicos que se pretenden alcanzar. También incluye esta sección de organización de la memoria para orientar al lector sobre la estructura del documento.

Capítulo 2 – Estado del arte: recoge un análisis comparativo de aplicaciones similares ya existentes, con el objetivo de identificar puntos fuertes y carencias que puedan aprovecharse como referencia para el desarrollo del proyecto. Además, se detallan las tecnologías empleadas, justificando su elección en función de las necesidades de la aplicación.

Capítulo 3 – Requisitos, especificaciones, costes, riesgos y viabilidad: define los requisitos funcionales y no funcionales del sistema, junto con sus especificaciones técnicas.

También se lleva a cabo una estimación de los costes asociados, se identifican posibles riesgos y se valora la viabilidad del proyecto desde una perspectiva técnica y económica.

Capítulo 4 – Análisis: describe el análisis de la solución propuesta, incluyendo aspectos como la estructura general de la aplicación, la lógica funcional y la interacción entre sus componentes principales.

Capítulo 5 – Diseño: presenta el diseño de la aplicación a diferentes niveles (arquitectura, interfaz de usuario, estructura de datos, etc.), sirviendo como puente entre el análisis conceptual y la posterior implementación técnica.

Capítulo 6 – Implementación y pruebas: documenta el proceso de desarrollo, indicando cómo se ha llevado a cabo la implementación de la solución y qué herramientas se han utilizado. A su vez, se incluyen las pruebas realizadas para verificar el correcto funcionamiento del sistema, abarcando pruebas funcionales, de rendimiento y de usabilidad.

Capítulo 7 – Conclusiones: ofrece una reflexión final sobre los resultados obtenidos, revisa los costes reales frente a los estimados, valora el cumplimiento de los objetivos y propone posibles mejoras o líneas de trabajo futuro.

Bibliografía: recoge todas las fuentes utilizadas para la elaboración del trabajo, tanto en su fase de investigación como en el desarrollo e implementación del proyecto.

Esta organización busca facilitar la lectura y comprensión del documento, guiando al lector a través del proceso completo de conceptualización, desarrollo y evaluación de la aplicación DONA'm MÓN.

Capítulo 2

Estado del arte

2.1. Historia, evolución e impacto de las aplicaciones móviles

2.1.1. Origen y evolución de los dispositivos móviles

El desarrollo de las aplicaciones móviles no puede entenderse sin revisar primero la evolución histórica de los dispositivos que las alojan. La telefonía móvil nace en 1946 con el sistema móvil de AT&T, que permitía realizar llamadas desde vehículos utilizando dispositivos de más de 30 kg. Este sistema inicial, basado en la tecnología Push-To-Talk y gestionado por operadoras humanas, tenía una cobertura muy limitada y altos costes de uso [3].

Durante las décadas siguientes se sucedieron avances importantes: la introducción del sistema RCC en 1960, el Improved Mobile Telephone Service de AT&T en 1965, y finalmente, la llegada del primer teléfono móvil portátil, el Motorola DynaTAC, en 1983. Sin embargo, no fue hasta la década de 1990 cuando los dispositivos comenzaron a incorporar funcionalidades propias de asistentes personales, como los Apple Newton o los Palm Pilot, marcando así el inicio de las primeras “aplicaciones” de uso cotidiano, como agendas, calendarios y juegos como Snake de Nokia [4].

El cambio de paradigma llegó en 2007 con la aparición del iPhone, que introdujo una interfaz multitáctil intuitiva y eliminó el teclado físico, inaugurando así la era del smartphone moderno. Un año más tarde, Apple lanzó la App Store, permitiendo a desarrolladores externos distribuir aplicaciones fácilmente. Ese mismo año, Google presentó Android junto con el HTC Dream, diversificando el ecosistema y facilitando el acceso masivo al desarrollo y consumo de aplicaciones móviles [5].



Figura 2.1: Evolución histórica de los dispositivos móviles.

2.1.2. Ecosistemas y sistemas operativos móviles

El auge de las aplicaciones móviles está estrechamente vinculado al desarrollo de los sistemas operativos (SO) móviles. Durante los primeros años del siglo XXI, Symbian OS dominaba el mercado, especialmente en terminales de Nokia. Sin embargo, fue rápidamente desplazado por iOS (2007) y Android (2008), que ofrecían interfaces gráficas avanzadas, tiendas de aplicaciones centralizadas y acceso optimizado a hardware [6].

Actualmente, Android posee una cuota de mercado superior al 70 % a nivel global, mientras que iOS concentra cerca del 28 % del mercado, especialmente en países desarrollados [7]. Otros sistemas operativos como Windows Phone, BlackBerry OS o Tizen han desaparecido o mantienen una presencia marginal. La consolidación de este duopolio ha propiciado estándares de desarrollo bien definidos, facilitando la expansión del mercado de aplicaciones.

Cabe destacar que la fragmentación del ecosistema Android sigue representando un reto técnico. Según un estudio de Android Police [8], el tiempo medio de soporte de actualizaciones es de 21 meses en dispositivos Android, frente a los 41 meses de Apple, lo que obliga a los desarrolladores a implementar múltiples versiones y adaptaciones para garantizar la compatibilidad.

2.1.3. Tipos de aplicaciones móviles

Las aplicaciones móviles pueden clasificarse en función de su arquitectura en cinco categorías principales:

- **Aplicaciones web:** ejecutadas dentro del navegador, sin necesidad de instalación. Utilizan tecnologías como HTML5, CSS y JavaScript. Son portables pero con acceso limitado al hardware [?].

- **Aplicaciones web progresivas (PWA)**: ofrecen una experiencia cercana a las nativas, pueden instalarse desde el navegador y ejecutarse sin conexión, gracias a tecnologías como los service workers.
- **Aplicaciones híbridas**: encapsulan una web app dentro de una aplicación nativa mediante motores como Cordova o Ionic. Permiten cierto acceso a funciones del dispositivo, pero con rendimiento inferior a las apps nativas.
- **Aplicaciones nativas**: desarrolladas específicamente para cada plataforma, usando lenguajes como Swift (iOS), Kotlin o Java (Android). Ofrecen el mejor rendimiento, pero requieren múltiples desarrollos independientes.
- **Aplicaciones nativas multiplataforma**: permiten usar un único lenguaje y base de código para generar versiones para iOS y Android. Ejemplos incluyen React Native, Flutter y Xamarin, ampliamente utilizados en el desarrollo moderno [9].

	App Nativa	App Híbrida	App Web (Progresiva)	App Web
Experiencia de usuario	EXCELENTE	EXCELENTE	BUENA	NORMAL
Velocidad y Rendimiento	MUY RÁPIDA	MUY RÁPIDA	BUENA	NORMAL
Seguridad	ALTA	ALTA	BUENA	BUENA
App Stores	SÍ	SÍ	NO	NO
Función Offline	SÍ	SÍ	NO	NO
Tiempo de Desarrollo	ALTO	MEDIO	BAJO	BAJO
Mantenimiento	ALTO	MEDIO	BAJO	BAJO
Coste de Desarrollo	ALTO	MEDIO	MEDIO	MEDIO

Cuadro 2.1: Comparación entre tipos de aplicaciones móviles

Cada enfoque presenta ventajas y limitaciones en términos de rendimiento, coste, mantenimiento, compatibilidad y acceso a recursos del sistema.

2.1.4. Impacto económico y social

El mercado de aplicaciones móviles ha alcanzado dimensiones económicas y sociales extraordinarias. En 2023 se descargaron más de 257 mil millones de aplicaciones a nivel

mundial, con un gasto en apps y suscripciones que superó los 170 mil millones de dólares [10]. El tiempo medio diario de uso del móvil ha aumentado a 5 horas, consolidando a las apps como principales mediadoras de nuestra relación con el entorno digital [11].

Además, su impacto trasciende lo económico. Las apps han revolucionado sectores clave como la salud (mHealth), la educación (mLearning), el transporte, el comercio electrónico, la cultura y la participación ciudadana. En el ámbito educativo, permiten estrategias de aprendizaje ubicuo y contextualizado, especialmente relevantes para iniciativas como la desarrollada en este trabajo, que integra realidad aumentada y geolocalización para visibilizar el patrimonio invisible en la ciudad.

En cuanto a la accesibilidad, los dispositivos móviles permiten una conexión constante con contenidos digitales, lo que favorece procesos de inclusión digital en poblaciones diversas. Esta capacidad de mediación sociotécnica convierte a las aplicaciones móviles en herramientas clave para la innovación social [?].

2.2. Evolución de la Realidad Aumentada en el Contexto de las Aplicaciones Móviles

La Realidad Aumentada (RA) es una tecnología que permite combinar contenido digital generado por ordenador con la percepción del mundo físico, proporcionando una experiencia enriquecida y contextual. Aunque su auge reciente está vinculado al ecosistema móvil, su desarrollo abarca más de medio siglo, con avances teóricos, técnicos y aplicados que han configurado su estado actual como tecnología madura.

2.2.1. Fundamentos y primeras ideas (1960–1990)

Los antecedentes de la RA preceden a la era digital. En 1962, Morton Heilig presentó el Sensorama, un sistema inmersivo que combinaba estímulos visuales, sonoros y olfativos para simular experiencias sensoriales integradas, aunque sin capacidades de interacción. A nivel tecnológico, el verdadero precursor de la RA fue el Head-Mounted Display (HMD) diseñado por Ivan Sutherland en 1968, apodado "Sword of Damocles". Este sistema óptico-mecánico, con seis grados de libertad, permitió visualizar objetos generados por ordenador superpuestos al entorno real [12].



Figura 2.2: Primer sistema RA: "Sword of Damocles" (1968)

No obstante, el concepto formal de RA se desarrolló más tarde. En 1992, Caudell y

Mizell acuñaron el término Augmented Reality en un contexto industrial, para referirse al uso de displays que asistían a operarios durante tareas complejas. Este enfoque pragmático orientó la investigación hacia sistemas útiles para tareas del mundo real, diferenciándose de la realidad virtual (RV), que propone la inmersión total en entornos digitales.

2.2.2. Definición formal y primeros prototipos (1990–2000)

Ronald Azuma estableció en 1997 la definición académica más citada de RA, basada en tres criterios fundamentales: (1) combinación de elementos reales y virtuales, (2) interacción en tiempo real y (3) registro espacial tridimensional preciso [12]. En paralelo, Paul Milgram y Fumio Kishino introdujeron el Continuo Realidad-Virtualidad, que posiciona la RA entre el entorno físico puro y la RV completa, permitiendo caracterizar distintos grados de mezcla entre ambos mundos [13].



Figura 2.3: Sistema KARMA para asistencia con RA (1993)

Durante esta década se desarrollaron los primeros sistemas funcionales de RA. En 1993, el sistema KARMA, creado por Steve Feiner y su equipo, permitía la superposición de información contextual durante tareas físicas [14]. En 1997, se presentó el prototipo MARS, un sistema de RA móvil diseñado para guiar a turistas mediante visualización aumentada en tiempo real. Estas iniciativas fueron posibles gracias al desarrollo de dispositivos portátiles y técnicas de tracking visual.

2.2.3. Herramientas abiertas y salto a lo portátil (1999–2010)

Uno de los hitos más relevantes en esta etapa fue el desarrollo de ARToolKit por Hirokazu Kato y Mark Billinghurst en 1999. Esta librería de código abierto permitió detectar marcadores visuales en video y superponer contenido digital 3D en tiempo real [15]. Supuso una democratización de la RA, facilitando la experimentación por parte de investigadores y desarrolladores independientes.

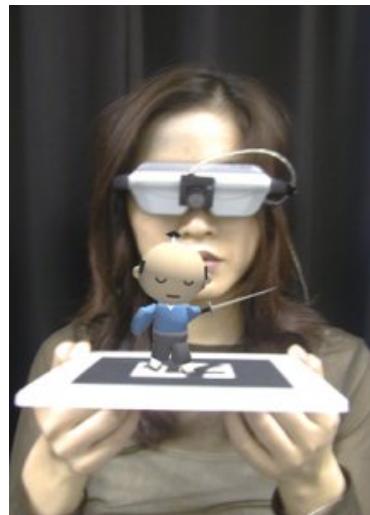


Figura 2.4: ARToolKit y marcadores visuales (1999)

Entre 2001 y 2004, surgieron los primeros sistemas de RA sobre dispositivos móviles, como PDAs y teléfonos inteligentes de primera generación. En 2003, Wagner y Schmalstieg presentaron un sistema de RA completamente autónomo en un dispositivo de mano [16]. Otros experimentos destacados fueron ARQuake (2000), un videojuego de RA en primera persona, y Invisible Train (2004), una aplicación colaborativa con varios usuarios simultáneos.

2.2.4. Consolidación y tendencias actuales (2010–2025)

Durante la última década, la Realidad Aumentada ha pasado de ser una tecnología emergente a consolidarse como una herramienta madura y transversal en múltiples sectores. El desarrollo de kits como ARToolKit, ARCore (Google) y ARKit (Apple) permitió la integración de funcionalidades de RA en dispositivos móviles de forma nativa, facilitando su adopción masiva y el desarrollo de aplicaciones accesibles sin necesidad de hardware especializado [17, 18].

Uno de los catalizadores clave de esta expansión fue el éxito comercial de Pokémon GO (2016), que demostró el potencial de la RA móvil como forma de interacción lúdica y geolocalizada [19]. Paralelamente, surgieron dispositivos avanzados como Microsoft HoloLens y Magic Leap One, que ofrecieron experiencias inmersivas mediante visores ópticos, reconocimiento espacial y control gestual [20, 21].

En el plano técnico, los avances en visión por computador y en algoritmos de localización y mapeo simultáneo (SLAM) mejoraron notablemente la precisión del registro espacial tridimensional [22]. También se consolidaron nuevas formas de interacción como la RA tangible y la RA colaborativa, que permiten manipular objetos físicos para modificar contenidos digitales o interactuar con varios usuarios en tiempo real [23, 24].

Actualmente, la RA se aplica con éxito en campos como la educación, la salud, la industria, la arquitectura o el comercio, y su desarrollo se orienta hacia una fusión más natural entre el mundo físico y el digital mediante técnicas de spatial computing y nuevos dispositivos como el Apple Vision Pro (2024) [25].



Figura 2.5: HoloLens 2 de Microsoft en labores de asistencia remota y formación.

2.3. La visibilización de las mujeres en el espacio urbano: una cuestión de justicia histórica

La representación de las mujeres en el espacio urbano continúa siendo escasa. Según el proyecto europeo “Women’s Legacy” [26], impulsado desde la Conselleria d’Educació, Cultura i Esport de la Generalitat Valenciana, solo un 10 % de las calles con nombres de personas en muchas ciudades españolas están dedicadas a mujeres, y este patrón se repite en monumentos, placas conmemorativas y espacios simbólicos.

A pesar de esta disparidad, en los últimos años han surgido iniciativas locales e institucionales orientadas a revertir esta situación. En Valencia, destaca el proyecto “Dones de Ciència” [27], una colaboración entre la Universitat Politècnica de València (UPV) y Las Naves, que ha creado más de 30 murales urbanos en centros educativos para homenajear a mujeres científicas de todo el mundo. Esta intervención artística no solo embellece el entorno, sino que actúa como una herramienta pedagógica para inspirar a las nuevas generaciones. Asimismo, el Ayuntamiento de València ha promovido campañas de renombramiento de calles con nombres femeninos y la inclusión de mujeres en los itinerarios culturales y turísticos de la ciudad [28].

Por otra parte, diversas asociaciones vecinales y colectivos feministas han impulsado rutas autogestionadas por la ciudad que visibilizan las huellas femeninas en el espacio urbano. Entre ellas destaca la Ruta Violeta de Mujeres en Valencia”, organizada por la Asociación Por Ti Mujer, que recorre espacios emblemáticos ligados a la historia de las mujeres en la ciudad, promoviendo un reconocimiento público de sus contribuciones sociales y culturales [29].

Asimismo, la Assemblea Feminista de València ha organizado la Ruta pels espais del Patronato de Protección a la Mujer”, que recorre espacios vinculados a esta institución franquista, la cual operó durante gran parte del siglo XX bajo la tutela de la Iglesia y el Estado. Su función no era tanto proteger a mujeres víctimas de violencia como controlar y castigar a aquellas que transgredían los roles tradicionales de género. Esta ruta permite evidenciar cómo las estructuras institucionales legitimaron la exclusión, la represión y la estigmatización de mujeres consideradas “desviadas” por la moral conservadora, visibilizando así dinámicas de resistencia y memoria colectiva en la ciudad [30].

Además, la Ruta de Memorias Lesbianas”, también organizada por la Assemblea Feminista de València, recorre lugares significativos para la comunidad lesbica, con el objetivo

de visibilizar los derechos y experiencias de las mujeres lesbianas en la Valencia de los años 70, 80 y 90, destacando la importancia de reconocer y valorar estas luchas en la construcción de la historia urbana [31].

Estas rutas permiten redescubrir barrios desde una óptica feminista, incorporando historias personales, luchas sociales y legados invisibilizados. En este sentido, la plataforma DONA'm MÓN podría ser un recurso útil a implementar durante estas rutas, integrando información geolocalizada, recursos multimedia y facilitando la participación activa de las personas en la construcción de esta memoria feminista urbana.

2.3.1. Referentes femeninos en la ciudad: biografías y puntos de interés

La aplicación propuesta busca poner en valor la figura de mujeres que, desde distintos campos del saber y la acción, han contribuido de manera decisiva al progreso de los derechos, el pensamiento y la cultura. Aunque muchas de ellas han nacido o trabajado en Valencia, su legado trasciende el ámbito local y constituye una aportación fundamental a la historia del feminismo y la igualdad. A través de la geolocalización de lugares clave asociados a sus vidas y obras, se propone una experiencia que va más allá de la información enciclopédica, para convertirse en un ejercicio de reconocimiento social y emocional de su presencia en la ciudad.

Una de las figuras destacadas es Concepción Aleixandre Ballester (1862–1952), una de las primeras mujeres médicas en España y pionera en el ámbito de la ginecología. Su activismo feminista se centró en la defensa del acceso de las mujeres a la educación y al ejercicio profesional de la medicina, algo profundamente innovador en su tiempo. Estudió en la Universidad de Valencia, donde enfrentó obstáculos institucionales por su condición de mujer, y desarrolló una intensa carrera médica y científica. El edificio histórico de la Facultad de Medicina de la Universitat de València es un lugar simbólico para recordar su figura y su lucha por la igualdad en el ámbito sanitario [32].

Otra mujer fundamental en la historia reciente de Valencia es Carmen Alborch (1947–2018). Fue ministra de Cultura, escritora, catedrática de Derecho Mercantil y una de las voces más influyentes del feminismo institucional en España. Su obra literaria, como *Solas* (1999), promovió un feminismo humanista accesible y empático. En Valencia, dejó una impronta profunda como directora del Instituto Valenciano de Arte Moderno (IVAM), donde impulsó una política cultural abierta, plural y comprometida con la igualdad. El IVAM, por tanto, no solo representa su labor como gestora cultural, sino también su apuesta por un feminismo integrador desde las instituciones [33].

En el ámbito de los derechos civiles, Clara Campoamor (1888–1972) ocupa un lugar esencial por su incansable lucha por el sufragio femenino, que logró defender con éxito en las Cortes Constituyentes de 1931. Si bien no tuvo una vinculación directa con Valencia, su legado ideológico inspiró a generaciones de mujeres en toda España, incluida la Comunidad Valenciana. La existencia de centros educativos como el CEIP Clara Campoamor subraya la importancia de su figura como referente pedagógico y su impacto en la formación de una conciencia feminista desde edades tempranas [34].

Retrocediendo a la Edad Media, encontramos a Isabel de Villena (1430–1490), abadesa del Real Monasterio de la Trinidad y considerada la primera escritora en lengua valenciana. Su obra *Vita Christi* constituye una relectura humanista del relato evangélico desde una perspectiva femenina, resaltando la figura de la Virgen y otras mujeres bíblicas. Isabel de Villena transformó el espacio monástico en un lugar de producción intelectual y espiritual

liderado por mujeres, algo excepcional en su época. El Monasterio de la Trinidad, donde vivió y escribió, es un lugar clave para reivindicar la aportación femenina a la literatura y la espiritualidad valencianas [35].

En la medicina, Manuela Solís Clarás (1888–1936) destaca por ser una de las primeras mujeres médicas de la ciudad. Ejerció en el Hospital Provincial de Valencia y fue una firme defensora del acceso de las mujeres a la formación sanitaria. Además de su práctica médica, se implicó en la divulgación científica y en la promoción de redes de apoyo entre profesionales sanitarias, en un momento en que la medicina era un espacio profundamente masculinizado [36]. El Hospital Provincial es, por tanto, un espacio idóneo para recordar su legado.

La aplicación también incluye representaciones artísticas contemporáneas de mujeres de gran relevancia internacional. Por ejemplo, Mae Jemison, primera mujer afroamericana en viajar al espacio, es homenajeada con un mural en el CEIP Torrefiel. Su figura simboliza la superación de barreras raciales y de género en los campos de la ciencia y la exploración espacial. Del mismo modo, Vandana Shiva, ecofeminista india y activista medioambiental, cuenta con un mural en el CEIP Ballester Fandos. Su lucha por la justicia ambiental, la soberanía alimentaria y los derechos de las mujeres rurales conecta con un feminismo global que también debe tener presencia en las narrativas educativas locales [37].

“DONA’m MÓN” busca ser un vehículo para visibilizar la presencia de estas figuras en el espacio urbano, contribuyendo así a una representación más justa y equitativa de la historia y promoviendo un futuro en el que la diversidad y la igualdad sean valores fundamentales.

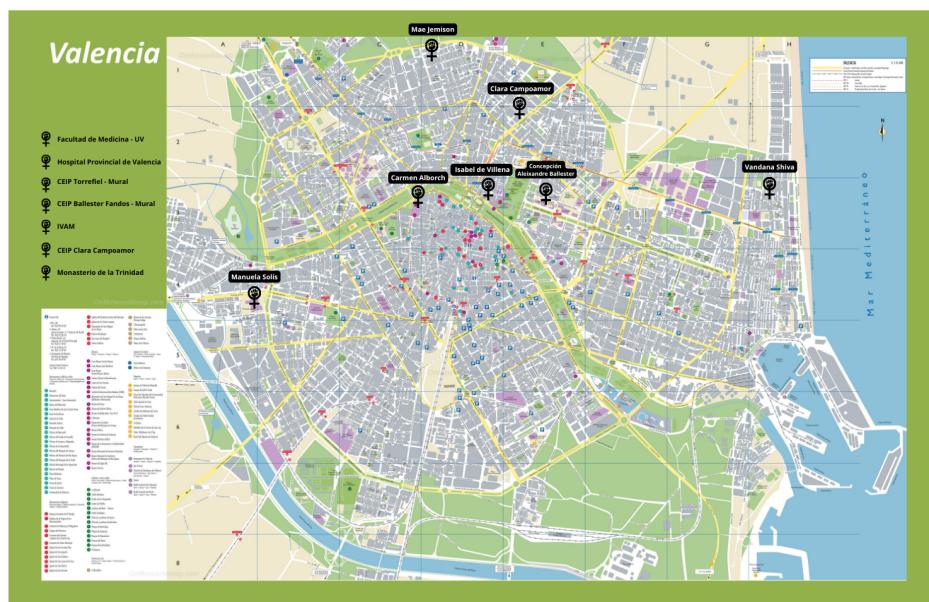


Figura 2.6: Mapa de Valencia con los puntos geolocalizados de las mujeres referentes incluidos en la aplicación.

2.4. Análisis de aplicaciones similares

En el desarrollo de “DONA’m MÓN”, resulta esencial analizar aplicaciones existentes que, de manera similar, combinan geolocalización y realidad aumentada (RA) para ofrecer experiencias inmersivas y educativas. Este análisis nos permite identificar características, estrategias y tecnologías clave que podrían ser adaptadas para cumplir con los objetivos específicos del proyecto. A continuación, se presentan algunas aplicaciones relevantes cuyos enfoques, funcionalidades y tecnologías han inspirado aspectos concretos del diseño conceptual de este TFG.

2.4.1. PokéMon GO

Una de las referencias más notables es PokéMon GO, que revolucionó el mercado de aplicaciones móviles al combinar geolocalización y RA para motivar a los usuarios a explorar el mundo real. En este caso, los jugadores buscan y capturan criaturas virtuales que aparecen en puntos geográficos específicos. La aplicación también implementa un sistema de recompensas y niveles que fomenta la participación constante. Aunque el enfoque principal de PokéMon GO es el entretenimiento, su éxito demuestra cómo las tecnologías inmersivas pueden transformar la forma en que las personas interactúan con su entorno. En “DONA’m MÓN”, se busca adaptar esta dinámica de exploración y descubrimiento, pero con un enfoque educativo y cultural, utilizando los puntos de interés como portales hacia las historias de mujeres que dejaron una huella significativa en Valencia.



Figura 2.7: Interfaz de PokéMon GO mostrando un entorno real aumentado con la aparición de un PokéMon virtual y también el entorno virtual del juego.

2.4.2. Streetmuseum

Por otro lado, Streetmuseum es una aplicación que también aprovecha la RA, pero con un objetivo más centrado en la historia y la cultura. Esta herramienta permite a los usuarios visualizar imágenes y escenas históricas superpuestas al paisaje actual, ofreciendo una perspectiva del pasado en tiempo real. La capacidad de conectar visualmente el entorno moderno con eventos y figuras históricas es una inspiración directa para el proyecto. En nuestro proyecto, esta idea se adapta al permitir que los usuarios accedan a contenido multimedia –como imágenes, textos y modelos 3D– que contextualice la vida y obra de las mujeres en cada ubicación geolocalizada.



Figura 2.8: Ejemplo de Streetmuseum: visualización de una escena histórica superpuesta sobre el entorno actual de la ciudad.

2.4.3. CultuAR

CultuAR, desarrollada por la empresa española AR Vision, es una aplicación que combina la geolocalización con la realidad aumentada para ofrecer visitas culturales interactivas en más de 200 municipios. Mediante la cámara del dispositivo, los usuarios pueden visualizar reconstrucciones históricas, personajes del pasado o información multimedia superpuesta al entorno real. Una de sus características más valiosas es la capacidad de personalizar rutas culturales por municipios, integrando elementos inmersivos que enriquecen la experiencia turística y patrimonial. Esta aplicación destaca por su integración fluida con el entorno urbano y su enfoque didáctico, lo que representa una clara inspiración para DONA'm MÓN en cuanto a accesibilidad y diseño de experiencias geolocalizadas con RA.



Figura 2.9: Captura de CultuAR mostrando información sobre un monumento en el entorno real.

2.4.4. Historypin

Finalmente, Historypin se presenta como una plataforma colaborativa que permite a los usuarios subir y explorar fotografías, eventos y recuerdos históricos asociados a ubicaciones específicas. Lo más destacable de esta aplicación es su enfoque participativo, donde los usuarios pueden contribuir activamente al contenido disponible. Este modelo de colaboración inspira la posibilidad de que, en futuras versiones de “DONA'm MÓN”, los usuarios también puedan agregar información o enriquecer las narrativas existentes sobre mujeres de Valencia, convirtiendo la aplicación en una herramienta dinámica y viva.

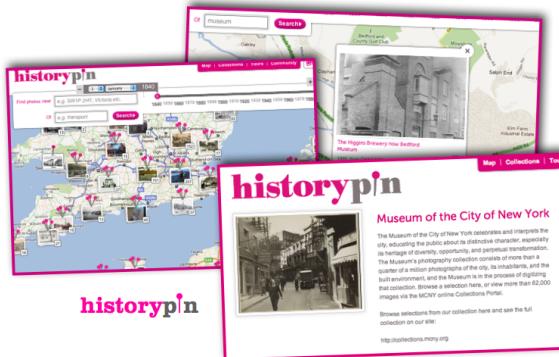


Figura 2.10: Interfaz de Historypin donde se visualizan fotografías históricas asociadas a ubicaciones reales en el mapa y su detalle de contenido.

Estas cuatro aplicaciones ilustran diversas maneras de utilizar la tecnología para crear experiencias inmersivas e interactivas. Cada una de ellas presenta fortalezas únicas que se alinean, en mayor o menor medida, con los objetivos de DONA'm MÓN. De Pokémon GO tomamos la dinámica de exploración y gamificación; de Streetmuseum, la idea de superponer contenido histórico en el entorno actual; de CultuAR, el uso efectivo de RA para la divulgación cultural local; y de Historypin, el potencial de colaboración comunitaria. Esta combinación de estrategias y conceptos pretende hacer de DONA'm MÓN una aplicación única que conecte a los usuarios con la historia de manera inclusiva, educativa e innovadora.

2.5. Tecnologías

En este apartado se analizan las tecnologías disponibles y las soluciones existentes en el mercado que pueden ser aplicadas al desarrollo de la aplicación móvil. Este análisis incluye una revisión de herramientas de geolocalización, realidad aumentada, bases de datos y frameworks de desarrollo multiplataforma. La selección de las tecnologías se justifica considerando la funcionalidad, compatibilidad y eficiencia necesarias para cumplir con los objetivos del proyecto.

2.5.1. Plataformas de desarrollo de aplicaciones móviles

Una de las decisiones fundamentales en cualquier proyecto de ingeniería de software es la elección de la tecnología base sobre la cual se construirá la aplicación. Esta decisión afecta directamente a la arquitectura general, la experiencia de usuario, la mantenibilidad del código y la escalabilidad futura del sistema. En el contexto de este proyecto, se evaluaron varias opciones para el desarrollo de una aplicación móvil multiplataforma que integrase geolocalización, funcionalidades de realidad aumentada, y una interfaz moderna, accesible y responsive.

Unity. Unity es un motor de desarrollo multiplataforma ampliamente utilizado en los ámbitos del videojuego, simulación y experiencias interactivas en tiempo real. Su potencia radica en su motor gráfico, que permite renderizar contenido 3D de alta calidad, así como en su flexibilidad para implementar lógica de aplicación mediante scripts en C#. Unity ofrece soporte directo para tecnologías de realidad aumentada a través de

módulos como AR Foundation, que actúan como capa de abstracción sobre ARKit (iOS) y ARCore (Android), permitiendo así desarrollar una única base de código RA para múltiples plataformas.

En lo relativo a interfaces de usuario, Unity ha evolucionado en los últimos años. Su sistema tradicional basado en el Canvas permitía crear interfaces gráficas bidimensionales, pero con limitaciones de estilo y escalabilidad. Como respuesta a estas deficiencias, se introdujo UI Toolkit, un nuevo sistema de interfaz basado en un modelo declarativo, inspirado en tecnologías web como HTML y CSS. UI Toolkit permite separar la lógica de presentación del comportamiento, definiendo componentes visuales reutilizables y estilos en archivos dedicados. A pesar de estas mejoras, el desarrollo de interfaces ricas, formularios interactivos y navegación compleja continúa siendo menos eficiente que en frameworks específicos para aplicaciones móviles.

React Native. React Native es un framework desarrollado por Meta (anteriormente Facebook) para el desarrollo de aplicaciones móviles multiplataforma utilizando JavaScript y el modelo de componentes de React. Permite crear interfaces de usuario nativas reutilizando componentes declarativos que se renderizan mediante puentes hacia los componentes nativos de Android e iOS. Una de las grandes ventajas de React Native es su arquitectura reactiva, que permite gestionar estados complejos de forma eficiente, manteniendo la UI sincronizada con los datos de la aplicación.

En combinación con Expo, un conjunto de herramientas que facilita la creación y prueba de aplicaciones React Native, es posible desarrollar aplicaciones sin necesidad de compilar ni configurar código nativo. Esto permite utilizar herramientas como Expo Go para probar la aplicación directamente en un dispositivo físico mediante código QR, lo que acelera significativamente el flujo de desarrollo. React Native es especialmente potente para construir interfaces modernas, personalizadas y adaptables, con soporte para navegación compleja, formularios, animaciones y estilos dinámicos.

Flutter. Flutter es un framework desarrollado por Google para construir aplicaciones multiplataforma desde una única base de código. Utiliza el lenguaje Dart, y su arquitectura se basa en un sistema de renderizado propio que dibuja todos los elementos de la UI desde cero, lo que garantiza uniformidad visual entre plataformas. Flutter ofrece una amplia colección de widgets personalizables y anima la creación de interfaces complejas de forma declarativa y eficiente.

Aunque Flutter no fue la opción elegida para este proyecto, se considera una alternativa competitiva frente a React Native. Sin embargo, su adopción requiere dominar el lenguaje Dart, y su integración con tecnologías de realidad aumentada aún está en evolución en comparación con la madurez del ecosistema de Unity o la compatibilidad simplificada que ofrece React Native con herramientas web como AR.js y A-Frame.

Desarrollo de interfaz de usuario y elección final. Una parte crucial del proyecto es la interfaz gráfica de usuario, dado que la aplicación está destinada a un público general y debe facilitar la navegación intuitiva, la visualización de datos geográficos y el acceso a contenidos de realidad aumentada. Si bien Unity, especialmente con UI Toolkit, ofrece un entorno funcional para construir interfaces gráficas, su enfoque está más orientado a interacciones en entornos tridimensionales y menos a la gestión de flujos de navegación o formularios comunes en aplicaciones móviles.

Por el contrario, React Native destaca precisamente en este tipo de interfaz. Su modelo declarativo, su sistema de componentes reutilizables, su integración con herramientas modernas de desarrollo (como React Navigation, Redux o Context API), y su compatibi-

lidad con Expo, permiten construir aplicaciones móviles escalables y con una experiencia de usuario optimizada.

Por estos motivos, y tras un análisis técnico detallado, se decidió utilizar React Native con Expo como entorno principal de desarrollo para esta aplicación, reservando el uso de tecnologías gráficas (como la realidad aumentada) a componentes embebidos que se integran dentro de esta arquitectura general. Esta elección permite combinar la robustez de React para la UI con la flexibilidad de integrar experiencias multimedia mediante tecnologías web.

2.5.2. Geolocalización

La geolocalización es uno de los pilares fundamentales de la aplicación, ya que permite situar puntos de interés específicos en un mapa, facilitando que los usuarios exploren la ciudad de Valencia mientras descubren la historia de mujeres destacadas. Para implementar esta funcionalidad, existen diversas herramientas ampliamente utilizadas en el desarrollo de aplicaciones móviles:

- **Google Maps API:** Es una solución robusta y versátil que permite integrar mapas interactivos, obtener datos de ubicación en tiempo real y personalizar la visualización según las necesidades del proyecto. Ofrece documentación detallada y soporte técnico, lo que facilita su integración. Sin embargo, su principal desventaja es que su plan gratuito únicamente se ofrece durante un periodo de prueba de corta duración y su coste se incrementa en proyectos con un gran volumen de usuarios o solicitudes [38].
- **Mapbox:** Alternativa potente que combina funcionalidades avanzadas, un alto nivel de personalización estética y una política de precios más flexible que Google Maps API, ya que su plan gratuito no tiene un límite de tiempo. Es especialmente útil en aplicaciones que requieren un diseño visual distintivo. No obstante, su integración puede resultar más compleja para desarrolladores con poca experiencia [39].
- **OpenStreetMap (OSM):** Opción de código abierto que permite el acceso y uso gratuito de datos cartográficos, además de ofrecer un alto nivel de personalización. Esta plataforma colaborativa es especialmente adecuada para proyectos con presupuestos limitados o académicos, ya que no implica costes de licencia. Aunque carece de algunas funcionalidades avanzadas nativas, como el enrutamiento en tiempo real o servicios de geolocalización enriquecidos, su flexibilidad y libertad de uso la convierten en una solución funcional y accesible [40].
- **Mapas nativos del dispositivo con React Native:** Esta opción permite aprovechar las capacidades de geolocalización del propio sistema operativo del dispositivo móvil (iOS o Android) mediante bibliotecas como react-native-maps. Es una solución eficiente y ligera, que facilita la integración con el hardware del dispositivo y proporciona una experiencia fluida. Además, evita costes derivados de servicios externos, lo que la convierte en una alternativa atractiva para proyectos con recursos limitados [41].

En este proyecto se ha optado por utilizar mapas nativos con React Native, ya que permiten una integración directa con la aplicación móvil, ofrecen un rendimiento óptimo, y eliminan la dependencia de servicios externos de pago. Esta elección responde a criterios de eficiencia, sostenibilidad económica y control sobre el desarrollo.

2.5.3. Realidad Aumentada (RA)

Una vez definida la arquitectura general de la aplicación y confirmado el uso de React Native como entorno principal de desarrollo, fue necesario analizar detalladamente qué tecnologías de realidad aumentada (RA) podían integrarse de forma eficiente, respetando las limitaciones del ecosistema de desarrollo adoptado. Uno de los objetivos fundamentales era conservar la compatibilidad con Expo Go, evitando procesos de compilación nativa que dificultasen el flujo de desarrollo iterativo. Bajo esta premisa, se realizó un estudio de las distintas alternativas existentes, evaluando tanto su madurez técnica como su grado de integración con React Native.

Realidad aumentada con tecnologías nativas. Las soluciones nativas ofrecen, en general, el mayor grado de precisión, rendimiento y acceso a capacidades avanzadas del dispositivo. Entre estas destacan ARKit (para dispositivos iOS) y ARCore (para Android), los kits de desarrollo oficial proporcionados por Apple y Google, respectivamente. Ambas plataformas permiten el seguimiento del entorno, la detección de planos horizontales y verticales, la oclusión de objetos virtuales con respecto al entorno físico, el anclaje espacial y la iluminación adaptativa. Estas herramientas están diseñadas para crear experiencias inmersivas de alta calidad y son ampliamente utilizadas en entornos profesionales y comerciales.

Sin embargo, una de sus limitaciones fundamentales es su naturaleza monoplataforma. Para desarrollar una aplicación de RA que funcione tanto en Android como en iOS, sería necesario implementar dos versiones independientes, o bien utilizar un entorno unificador como Unity, que permite compilar una única base de código para múltiples sistemas operativos.

Unity, a través de su módulo AR Foundation, permite abstraer las diferencias entre ARKit y ARCore, y desarrollar experiencias RA multiplataforma con una arquitectura común. Unity es particularmente potente en términos de renderizado gráfico, gestión de modelos tridimensionales, simulación física y animación en tiempo real, por lo que constituye una de las herramientas más robustas para el desarrollo de RA avanzada.

No obstante, su integración con aplicaciones desarrolladas en React Native es compleja. El enfoque convencional implicaría desarrollar toda la aplicación en Unity, lo cual, como se ha discutido anteriormente, penaliza la usabilidad y escalabilidad de la interfaz. Por este motivo, se exploró también la posibilidad de utilizar Unity como una librería nativa embebida dentro de una aplicación React Native.

Unity como librería nativa integrada en React Native. Unity permite exportar un proyecto como una librería nativa —un archivo .aar para Android o .framework para iOS— que puede integrarse dentro de una aplicación móvil desarrollada con otras tecnologías, como React Native. Este enfoque, conocido como Unity as a Library, ofrece la posibilidad de desarrollar únicamente la parte de RA en Unity y exponerla como un módulo invocable desde la aplicación principal.

Mediante este patrón arquitectónico, es posible mantener la lógica general y la interfaz de usuario desarrolladas en React Native, y delegar en Unity únicamente el renderizado y gestión de la escena RA. La comunicación entre ambos entornos se realiza mediante puentes nativos que deben implementarse en Java/Kotlin (para Android) y Objective-C/Swift (para iOS).

Este enfoque ha sido demostrado en proyectos como react-native-unity, que ofrecen ejemplos funcionales de integración, así como en la documentación oficial de Unity (Unity

as a Library). No obstante, esta integración presenta múltiples desafíos técnicos:

- Es necesario ejectar el proyecto Expo, abandonando así la posibilidad de utilizar Expo Go y el flujo de desarrollo simplificado que ofrece.
- La configuración del proyecto requiere conocimientos avanzados en desarrollo nativo para cada plataforma.
- La integración y sincronización de eventos entre Unity y React Native debe implementarse manualmente.

Por tanto, aunque Unity como librería nativa representa una solución muy potente desde el punto de vista gráfico y funcional, su coste de integración y mantenimiento dentro de una arquitectura basada en React Native la convierte en una opción poco viable para este proyecto, cuyo foco está en la accesibilidad, rapidez de desarrollo y compatibilidad multiplataforma sin compilación nativa.

Realidad aumentada en React Native con tecnologías web. En busca de una alternativa más ligera y plenamente compatible con React Native y Expo, se optó por explorar las soluciones basadas en tecnologías web. Una de las más destacadas es AR.js, una librería de código abierto que permite implementar experiencias de RA directamente en navegadores móviles, sin necesidad de acceso a capas nativas. AR.js se basa en WebGL, Three.js y A-Frame, y permite implementar RA sobre marcadores visuales (marker-based AR) mediante la detección de patrones gráficos como Hiro, Kanji o diseños personalizados.

Complementando AR.js, el framework A-Frame facilita la construcción declarativa de escenas 3D y RA mediante HTML. Permite importar modelos .glb, añadir iluminación, cámaras, interacciones y animaciones sin necesidad de escribir código complejo de bajo nivel. Esta aproximación reduce significativamente la barrera de entrada al desarrollo de RA, y se adapta perfectamente al modelo de desarrollo web moderno.

Las escenas construidas con A-Frame y AR.js pueden ser desplegadas como aplicaciones web estáticas (por ejemplo, en GitHub Pages), e integradas dentro de la aplicación React Native utilizando el componente WebView, que actúa como un navegador embebido. Este enfoque permite mantener la lógica general y la interfaz de la aplicación dentro del ecosistema de React Native, mientras que la experiencia de RA se encapsula como una unidad independiente, reutilizable y fácilmente actualizable.

Viro React. Otra alternativa evaluada fue Viro React, un framework orientado a la creación de experiencias inmersivas en RA y realidad virtual directamente en React Native. Viro permite cargar modelos 3D, definir cámaras, aplicar animaciones, detectar superficies, y renderizar objetos virtuales en función del entorno físico. Su enfoque declarativo está alineado con la filosofía de React y ofrece un entorno de desarrollo intuitivo para experiencias visuales.

No obstante, Viro no es compatible con Expo Go, ya que depende de módulos nativos que requieren compilar la aplicación. Para utilizarlo, es necesario ejectar el proyecto y utilizar herramientas como EAS Build para generar versiones personalizadas de la app. Esta necesidad de compilación rompe el flujo de desarrollo ágil basado en Expo y contradice uno de los principios técnicos fundamentales del proyecto: evitar configuraciones nativas complejas para preservar la portabilidad y escalabilidad del sistema.

Elección final. Después de un análisis riguroso, se optó por utilizar AR.js combinado con A-Frame, integrados en la aplicación React Native mediante un componente WebView.

Esta solución permite cumplir los objetivos funcionales de la aplicación (visualización de modelos 3D sobre marcadores físicos), respetando al mismo tiempo los requisitos técnicos del entorno (compatibilidad con Expo, desarrollo ágil, y arquitectura limpia).

La elección de esta tecnología responde a criterios técnicos sólidos:

- Portabilidad total: la experiencia RA se ejecuta en el navegador y es accesible desde cualquier dispositivo moderno.
- Desacoplamiento arquitectónico: la escena RA es independiente del código base de la aplicación.
- Facilidad de mantenimiento: se pueden actualizar las escenas de RA sin necesidad de recompilar la app.
- Compatibilidad con Expo Go: se preserva el flujo de desarrollo iterativo y multiplataforma.

En resumen, esta arquitectura permite combinar las ventajas del desarrollo móvil moderno (mediante React Native) con la flexibilidad de tecnologías web para RA, consiguiendo una solución técnica equilibrada, funcionalmente completa y sostenible a largo plazo.

2.5.4. Base de Datos

2.5.5. Arquitectura de persistencia y gestión de datos

En el desarrollo de esta aplicación se ha optado por una arquitectura desacoplada que separa claramente el frontend y el backend. El frontend se implementa en React Native, lo que permite el despliegue multiplataforma en dispositivos móviles, mientras que el backend se desarrolla en Django, un framework web de alto nivel que facilita la creación de aplicaciones robustas y seguras [42].

Persistencia de datos en entornos relacionales y no relacionales

La persistencia de datos es un aspecto fundamental en el desarrollo de aplicaciones que requieren almacenar información de manera duradera. Existen dos enfoques principales para gestionar esta persistencia: las bases de datos relacionales, que estructuran la información en tablas con relaciones definidas mediante claves, y las bases de datos no relacionales, que permiten una mayor flexibilidad al almacenar los datos en formatos como documentos, grafos o pares clave-valor [43].

Las bases de datos relacionales son especialmente adecuadas cuando los datos presentan una estructura definida, relaciones entre entidades y necesidad de integridad referencial, mientras que las no relacionales ofrecen ventajas en contextos donde la escalabilidad horizontal, la alta disponibilidad o la variabilidad en los esquemas de datos son prioritarias [44].

Django y la arquitectura MVT

Django se basa en la arquitectura Modelo–Vista–Plantilla (MVT), un patrón similar al clásico MVC (Model–View–Controller), adaptado al ecosistema web de Python [45]. Este patrón organiza el código en tres componentes:

- **Modelo (Model):** Define la estructura de los datos y su comportamiento a través de clases Python, que se traducen automáticamente en tablas mediante el ORM (Object-Relational Mapping) de Django.
- **Vista (View):** Gestiona la lógica de negocio y responde a las solicitudes HTTP, procesando la información que llega desde el frontend.
- **Plantilla (Template):** Presenta la información al usuario utilizando HTML enriquecido con etiquetas dinámicas, aunque en este proyecto su papel es mínimo, ya que la interfaz se renderiza completamente en React Native.

La interacción entre Django y la base de datos está mediada por su ORM, lo que permite manipular los datos mediante objetos Python en lugar de consultas SQL explícitas. Esta capa de abstracción incrementa la seguridad —previniendo ataques como la inyección SQL— y facilita el mantenimiento del código [45].

Sistema de Gestión de Bases de Datos: MySQL

Para este proyecto se ha elegido MySQL como sistema de gestión de bases de datos relacional (SGBDR), debido a su madurez, rendimiento y plena compatibilidad con Django [46]. MySQL permite estructurar los datos de manera eficiente y soporta operaciones complejas con múltiples relaciones entre tablas, lo que resulta fundamental para gestionar tanto la información relacionada con las figuras históricas como los datos de geolocalización, multimedia y analíticas del uso de la app.

La configuración entre Django y MySQL se realiza mediante el uso de controladores compatibles (como `mysqlclient`), lo que asegura una integración fluida y un rendimiento óptimo.

Comparativa entre sistemas de bases de datos

A continuación se muestra una tabla comparativa entre los principales sistemas de gestión de bases de datos considerados:

Característica	SQLite	MySQL	MariaDB	MongoDB
Tipo	Relacional	Relacional	Relacional	No relacional
Escalabilidad	Baja	Alta	Alta	Muy alta
Rendimiento	Medio	Alto	Muy alto	Alto
Integridad referencial	Sí	Sí	Sí	No
Compatibilidad Django	Nativa	Completa	Completa	Parcial (con plugins)
Flexibilidad de esquema	Baja	Media	Media	Alta
Facilidad de configuración	Muy fácil	Media	Media	Media
Comunidad y soporte	Moderada	Muy activa	Activa	Muy activa

Cuadro 2.2: Comparativa entre SQLite, MySQL, MariaDB y MongoDB [43, 47, 48].

Consideraciones sobre otras tecnologías: MariaDB, SQLite, MongoDB y Cassandra

Aunque inicialmente se contempló el uso de MariaDB, una bifurcación de MySQL con mejoras en rendimiento y licencias completamente abiertas, finalmente se optó por MySQL por razones de compatibilidad institucional y soporte técnico [47]. SQLite, si bien está soportada nativamente por Django y resulta útil en fases de desarrollo o en aplicaciones de bajo requerimiento, no ofrece la escalabilidad ni la gestión de concurrencia necesarias para este proyecto [49].

En cuanto a las bases de datos no relacionales, cabe mencionar tecnologías como MongoDB y Cassandra. MongoDB, basada en documentos JSON, es ampliamente utilizada en aplicaciones que requieren estructuras de datos flexibles y cambios frecuentes en el esquema [48]. Cassandra, por su parte, está orientada a entornos distribuidos con grandes volúmenes de datos y alta disponibilidad [50]. Sin embargo, el enfoque relacional fue considerado más adecuado para las necesidades estructuradas y normalizadas del presente trabajo.

Capítulo 3

Requisitos, especificaciones, coste, riesgos, viabilidad

3.1. Requisitos

3.1.1. Requisitos funcionales

Los requisitos funcionales describen las acciones que el sistema debe poder realizar para satisfacer las necesidades del usuario y del administrador. En este proyecto, se centran en el uso de geolocalización, realidad aumentada, gestión de contenido y experiencia interactiva.

Lista de requisitos funcionales:

3.1.2. Requisitos funcionales

RF1: La aplicación debe permitir a los usuarios visualizar un mapa interactivo con los puntos geolocalizados de interés.

RF2: El sistema debe detectar la ubicación actual del usuario mediante GPS.

RF3: Al acercarse físicamente a un punto geolocalizado, la aplicación debe desbloquear el contenido asociado.

RF4: Cada punto geolocalizado debe estar vinculado a una mujer histórica, con contenido educativo multimedia.

RF5: La aplicación debe mostrar una ficha informativa con texto, imágenes y/o audio sobre cada figura femenina.

RF6: La app debe permitir visualizar elementos de realidad aumentada (como imágenes, objetos 3D o figuras animadas) al enfocar con la cámara.

RF7: La lista de puntos y el mapa deben permitir filtrar los lugares por el ámbito de investigación de las mujeres representadas.

RF8: El usuario debe poder ver qué puntos ha desbloqueado y cuáles no.

RF9: La aplicación debe permitir consultar la lista completa de mujeres disponibles con sus respectivas ubicaciones.

RF10: El usuario debe recibir una notificación cuando esté cerca de un punto no visitado.

- RF11: Debe existir un sistema de rutas que conecten diferentes puntos de interés temáticamente.
- RF12: Al pulsar sobre un punto en el mapa o la lista, debe abrirse su ficha detallada, desde la cual el usuario podrá marcar el lugar como visitado o no.
- RF13: El sistema debe permitir guardar el progreso del usuario (puntos visitados).
- RF14: El usuario debe disponer de un historial donde se registre la fecha y hora de cada punto visitado.
- RF15: La aplicación debe ser funcional tanto en dispositivos Android como iOS.
- RF16: Debe existir un panel de administración web para gestionar los puntos, contenidos y figuras históricas.
- RF17: El administrador debe poder añadir, editar o eliminar mujeres, ubicaciones y materiales multimedia desde el panel.
- RF18: El sistema debe permitir importar imágenes, vídeos y modelos 3D desde el panel de administración.
- RF19: El sistema debe validar que no se puede desbloquear un punto si el usuario no está físicamente cerca.
- RF20: La app debe ofrecer retroalimentación visual y sonora al desbloquear contenido.
- RF21: El sistema debe enviar datos al backend para mantener sincronizado el progreso del usuario.
- RF22: La aplicación debe cargarse en menos de 5 segundos desde su apertura inicial.
- RF23: La aplicación debe permitir a los usuarios registrarse y acceder mediante un sistema de inicio de sesión.
- RF24: Los usuarios deben poder iniciar sesión con un correo electrónico y contraseña.
- RF25: La aplicación debe mantener la sesión iniciada hasta que el usuario decida cerrarla.
- RF26: El usuario debe poder editar los datos de su perfil desde la aplicación.
- RF27: El administrador debe poder acceder, desde el panel de gestión, a la lista de usuarios registrados.
- RF28: El administrador debe poder visualizar el perfil de cada usuario, incluyendo su progreso (puntos visitados).
- RF29: El sistema debe permitir al administrador eliminar usuarios o restablecer sus datos si es necesario.
- RF30: Al escanear un código QR de un lugar, se debe actualizar el progreso de rutas temáticas dentro de la aplicación.
- RF31: El sistema debe permitir filtrar los puntos por estado de visita (visitados o no visitados).

RF32: Los filtros por estado de visita y por ámbito deben poder aplicarse de forma combinada.

RF33: El listado de puntos debe estar ordenado por cercanía al usuario.

RF34: La distancia en tiempo real entre el usuario y cada punto debe mostrarse junto a cada entrada del listado.

3.1.3. Requisitos no funcionales

Los requisitos no funcionales indican cómo debe comportarse el sistema más allá de lo que hace: establecen criterios de calidad, restricciones técnicas o normativas, y condiciones necesarias para su correcto desarrollo, despliegue y uso.

Requisitos no funcionales del producto

Estos requisitos definen las propiedades internas del sistema y sus comportamientos esperados, como rendimiento, usabilidad, seguridad o compatibilidad técnica.

RNF1: La aplicación debe cargarse completamente en menos de 5 segundos desde su apertura.

RNF2: El sistema debe ser capaz de gestionar al menos 100 usuarios simultáneos sin caída de rendimiento.

RNF3: Los modelos 3D deben estar optimizados para dispositivos móviles, con un tamaño inferior a 10 MB por unidad.

RNF4: El consumo de batería durante el uso continuo no debe superar el 10 % por hora.

RNF5: La interfaz debe ser intuitiva y permitir su uso sin formación previa.

RNF6: La tipografía y botones deben adaptarse automáticamente a diferentes resoluciones de pantalla.

RNF7: La aplicación debe ofrecer un modo accesible con texto ampliado y audio narrado.

RNF8: La navegación debe ser posible con una sola mano (diseño mobile-first).

RNF9: Compatible con Android 10+ y iOS 13+.

RNF10: El panel de administración debe ser accesible desde navegadores modernos (Chrome, Firefox, Safari, Edge).

RNF11: Las contraseñas de los usuarios deben almacenarse cifradas en la base de datos.

RNF12: La comunicación entre cliente y servidor debe estar cifrada mediante HTTPS.

RNF13: El backend debe estar desarrollado en Django y utilizar PostGIS (extensión de PostgreSQL para datos geoespaciales) como sistema gestor de bases de datos.

RNF14: La aplicación debe estar desarrollada con A-Frame, integrando geolocalización y RA mediante tecnologías WebXR y bibliotecas compatibles con A-Frame.

RNF15: La arquitectura debe ser modular y permitir añadir nuevos puntos o contenidos sin modificar el código base.

RNF16: Se debe estructurar el código y documentarlo adecuadamente para facilitar mantenimiento y ampliación.

Requisitos no funcionales de la organización

Estos requisitos están relacionados con decisiones internas del equipo desarrollador o institución que afectan a los métodos de trabajo, tecnologías permitidas o estructura del proyecto.

RNF17: El sistema debe permitir su despliegue en un servidor que soporte Django y PostGIS (PostgreSQL) según los medios disponibles en la UV.

RNF18: Debe evitarse el uso de servicios de alto coste por uso intensivo, priorizando soluciones open source cuando sea posible (por ejemplo, evitar cuotas de Google Maps si es viable usar OpenStreetMap).

RNF19: El panel de administración debe incluir autenticación de administradores con control de permisos.

RNF20: El proyecto debe desarrollarse siguiendo buenas prácticas de ingeniería del software: control de versiones, pruebas, documentación técnica y separación por capas.

Requisitos no funcionales externos

Son aquellos impuestos por normativas legales, estándares externos o políticas públicas que el sistema debe cumplir.

RNF21: El sistema debe cumplir con el Reglamento General de Protección de Datos (RGPD) en lo referente al tratamiento de datos personales.

RNF22: Los usuarios deben aceptar la política de privacidad y condiciones de uso antes de registrarse.

RNF23: El acceso a los datos de usuario debe estar restringido y protegido contra accesos no autorizados.

RNF24: El almacenamiento de datos personales (nombre, email, localización) debe hacerse de forma segura y limitada a lo estrictamente necesario.

3.2. Especificaciones

Una vez definidos los requisitos funcionales del sistema, es posible descomponerlos en funcionalidades concretas que formarán parte de la aplicación y su panel de gestión. A continuación se detalla el conjunto de acciones que los usuarios (tanto visitantes como administradores) podrán realizar.

Funcionalidades relacionadas con la geolocalización y el mapa

- F1. Visualizar un mapa interactivo con los puntos de interés distribuidos por Valencia.
- F2. Detectar en tiempo real la ubicación actual del usuario.
- F3. Mostrar puntos cercanos en función de la ubicación.

- F4. Notificar al usuario cuando se acerque a un punto no visitado.
- F5. Restringir el acceso al contenido si el usuario no está físicamente cerca del punto.

Funcionalidades educativas y de contenido

- F6. Acceder a la ficha de una mujer histórica al llegar a un punto.
- F7. Mostrar contenido multimedia asociado: texto, imágenes, audios o videos.
- F8. Activar realidad aumentada al enfocar con la cámara del dispositivo.
- F9. Visualizar objetos 3D o elementos históricos en RA.
- F10. Consultar un listado general de todas las mujeres incluidas en la app.
- F11. Reproducir contenido offline si ya ha sido descargado previamente.

Funcionalidades de interacción y gamificación

- F12. Desbloquear logros al visitar puntos o completar rutas temáticas.
- F13. Mostrar estadísticas personales (puntos visitados, logros obtenidos).
- F14. Ofrecer retroalimentación visual y sonora al desbloquear un nuevo punto.

Funcionalidades de usuario

- F15. Registrar un nuevo usuario mediante correo electrónico y contraseña.
- F16. Iniciar sesión en la aplicación.
- F17. Mantener la sesión activa entre usos.
- F18. Cerrar sesión manualmente.
- F19. Aceptar política de privacidad y condiciones de uso al registrarse.

Funcionalidades del panel de administración

- F20. Iniciar sesión como administrador desde el backend web.
- F21. Añadir nuevas figuras históricas con nombre, descripción y material multimedia.
- F22. Asociar cada mujer con un punto geolocalizado.
- F23. Cargar imágenes, audios, videos y modelos 3D desde el panel.
- F24. Editar o eliminar puntos o figuras ya creadas.
- F25. Ver la lista de usuarios registrados.
- F26. Consultar el progreso de cada usuario (puntos desbloqueados, logros).

F27. Eliminar usuarios si fuese necesario.

F28. Consultar métricas de uso: puntos más visitados, número de usuarios activos, etc.

Funcionalidades generales

F29. Adaptar la interfaz a distintos tamaños de pantalla.

F30. Activar modo accesible con texto grande y audio.

F31. Navegar por la app de forma intuitiva y con accesibilidad mobile-first.

F32. Garantizar el uso fluido y sin errores tanto en Android como en iOS.

3.3. Ciclo de vida del software

Para llevar a cabo el desarrollo del proyecto **DONA'm MÓN**, se ha adoptado el **modelo incremental**, una estrategia que permite construir el software en entregas funcionales sucesivas. Cada uno de estos incrementos representa un subconjunto del sistema completo que puede desarrollarse, evaluarse y refinarse de forma independiente, favoreciendo una mejora continua y una respuesta ágil ante cambios.

Ventajas del modelo incremental con prototipos

- **Adaptabilidad:** permite integrar cambios o mejoras frecuentes sin rehacer el sistema completo.
- **Validación progresiva:** permite obtener retroalimentación temprana y recurrente tras cada incremento funcional.
- **Reducción de riesgos:** al trabajar sobre versiones funcionales, se detectan errores antes de que escalen.
- **Mejora continua:** la calidad del producto se incrementa con cada iteración, sin necesidad de esperar al final para realizar pruebas.

Justificación de su elección

El proyecto consiste en una aplicación educativa con módulos interdependientes como realidad aumentada, geolocalización y escaneo QR. Estas funcionalidades requieren una validación continua y pueden experimentar ajustes según el diseño, pruebas con usuarios o cambios técnicos. Por ello, el modelo incremental permite desarrollar cada componente por separado, verificar su funcionamiento de forma aislada y luego integrarlo, asegurando una base estable en todo momento.

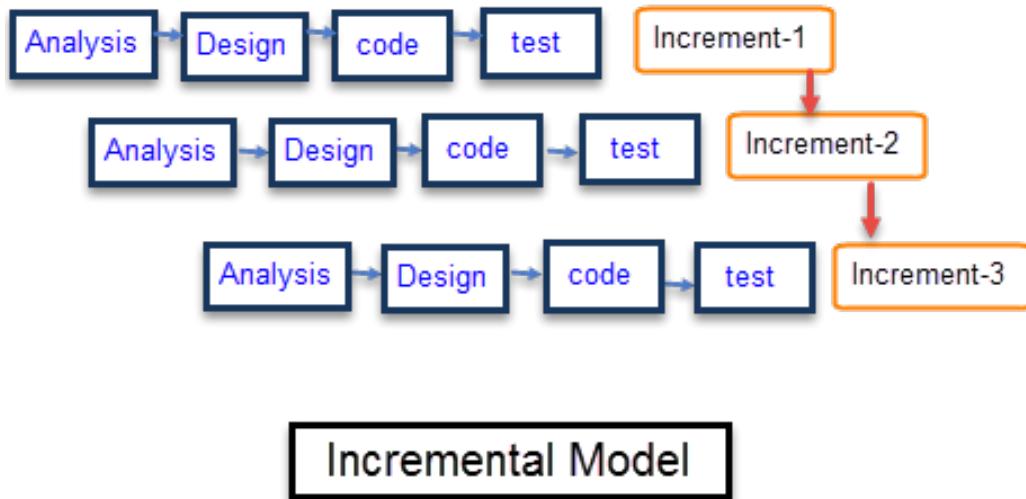


Figura 3.1: Representación del modelo incremental.

3.4. Planificación temporal

3.4.1. Descomposición de tareas por fases

La planificación se ha dividido en siete bloques principales: **Inicio**, **Estado del arte**, **Análisis**, **Diseño**, **Implementación**, **Pruebas** y **Documentación**, en concordancia con el modelo incremental utilizado.

1. Inicio

- Estudio del problema que resuelve la app
- Definición de objetivos del TFG
- Planificación inicial del proyecto

2. Estado del arte

- Revisión de apps similares educativas, con RA y mapas
- Estudio de mujeres y sus lugares en Valencia
- Estudio de frameworks de RA, mapas y escáneres QR
- Selección de tecnologías definitivas (React Native, Django, AFrame)

3. Análisis

- Definición detallada de requisitos funcionales y no funcionales
- Modelado inicial del sistema: navegación y casos de uso
- Modelado entidad-relación de la base de datos

- Definición de métricas y rutas

4. Diseño

- Diseño de la arquitectura frontend y backend
- Diseño de interfaz móvil y estilo visual
- Diseño de pantallas clave: login, listado, mapa, RA, etc.

5. Implementación backend

- Creación del proyecto Django y configuración Docker
- Creación de modelos de datos y migraciones
- Implementación de API REST para usuarios, lugares y rutas
- Sistema de login y sesiones
- Registro de visitas, logros y progreso

6. Implementación frontend

- Configuración base y navegación
- Registro e inicio de sesión de usuarios
- Implementación del listado y filtros por ámbito y visitas
- Implementación del historial de usuario
- Implementación de mapa con geolocalización
- Escaneo de QR y navegación a detalle
- Sincronización de datos con backend

7. Implementación de realidad aumentada

- Configuración de AR.js y marcador personalizado
- Activación de RA según la ubicación (menos de 1km)
- Visualización de modelo 3D y contenido RA educativo

8. Pruebas y validación

- Pruebas funcionales de cada módulo
- Pruebas completas con usuarios externos
- Corrección de errores y mejoras detectadas

9. Documentación

- Redacción progresiva de la memoria (paralela a todo el desarrollo)
- Revisión completa, maquetación y anexos finales

3.4.2. Estimación temporal

La técnica de estimación utilizada es la media beta de probabilidad. Para cada tarea se ha asignado un tiempo optimista (TO), pesimista (TP) y más probable (TM), y se ha calculado la duración estimada (TE) con la siguiente fórmula:

$$TE = \frac{TO+4\cdot TM+TP}{6}$$

Cuadro 3.1: Estimación por tres valores de las tareas del proyecto

Tarea	TO	TP	TM	TE
Estudio del problema	1	3	2	2.00
Definición de objetivos	1	2	1	1.17
Planificación inicial	1	2	1	1.17
Revisión apps similares	2	4	3	3.00
Estudio lugares mujeres	1	3	2	2.00
Frameworks RA y QR	2	4	3	3.00
Selección tecnologías	1	2	1	1.17
Requisitos funcionales	2	4	3	3.00
Casos de uso y navegación	1	3	2	2.00
Modelo E-R BD	1	3	2	2.00
Def. métricas y rutas	1	2	1	1.17
Diseño arquitectura	2	4	3	3.00
Diseño interfaz visual	2	5	3	3.17
Pantallas clave	2	4	3	3.00
Django y Docker	1	3	2	2.00
Modelos y migraciones	1	3	2	2.00
API usuarios y lugares	3	5	4	4.00
Login y sesiones	1	3	2	2.00
Registro visitas y logros	1	3	2	2.00
Conf. navegación base	1	3	2	2.00
Registro e inicio sesión	1	3	2	2.00
Listado y filtros	2	4	3	3.00
Historial de usuario	1	3	2	2.00
Mapa geolocalizado	2	4	3	3.00
Escaneo QR	1	3	2	2.00
Sincronización datos	1	3	2	2.00
Config. AR.js y marcador	1	3	2	2.00
Desbloqueo por distancia	1	3	2	2.00
Visualización RA	2	4	3	3.00
Pruebas funcionales	2	4	3	3.00
Pruebas con usuarios	1	3	2	2.00
Corrección errores	1	3	2	2.00
Redacción progresiva	4	6	5	5.00
Revisión y entrega final	2	4	3	3.00
Total estimado	73	127	96	100.00 días

3.4.3. Diagrama de Gantt

Para representar la planificación temporal del proyecto de forma visual, se ha elaborado un diagrama de Gantt utilizando Microsoft Project, teniendo en cuenta las dependencias y la posibilidad de solapar tareas como la documentación progresiva.



Figura 3.2: Diagrama de Gantt detallado del proyecto.

A pesar de que la estimación por tres valores del proyecto arroja una duración total de aproximadamente **100 días** (equivalente a **800 horas**), el desarrollo real se ha planificado a lo largo de **80 días laborables** (como puede observarse en el diagrama de Gantt), lo que supone un total de **640 horas efectivas** de trabajo, distribuidas en jornadas de **8 horas al día**.

Cabe destacar que:

- El proyecto se ha realizado en paralelo con el *Máster en Tecnologías Web, Computación en la Nube y Aplicaciones Móviles*.
- Además, durante este periodo se estaban llevando a cabo *prácticas extracurriculares en el Instituto de Robótica y Tecnologías de la Información y la Comunicación (IRTIC)*, donde también se ha avanzado el desarrollo del TFG.
- No se ha trabajado durante los fines de semana, por lo que los 80 días corresponden exclusivamente a días laborables.

La fecha de inicio del proyecto fue el **3 de marzo de 2025**, y la fecha de finalización fue el **20 de junio de 2025**. El proyecto ha sido planificado y ejecutado con el objetivo de ser presentado en la **primera convocatoria**.

3.5. Estimación de costes

La estimación económica del proyecto se ha realizado distinguiendo entre:

- **Costes directos de personal:** incluyen el salario bruto anual para cada perfil profesional, junto con los costes de Seguridad Social correspondientes (aproximadamente un 30 % adicionales). Se han considerado perfiles distintos (Jefe de Proyecto, Desarrollador Backend, Desarrollador Frontend, Desarrolador de Realidad Tester/-QA y Modelador 3D), aunque el proyecto sea llevado a cabo por una sola persona.
- **Costes directos de material:** contemplan licencias, equipos y amortización del hardware y software utilizado.
- **Costes indirectos:** coeficiente adicional del 20 % aplicado sobre el total de los costes directos, para cubrir gastos generales asociados (electricidad, internet, etc.).

Este desglose permite dimensionar correctamente los recursos implicados y simular una estructura empresarial, acorde con los requisitos formales del TFG.

3.5.1. Costes directos de personal

Se han obtenido los sueldos medios anuales en España para cada perfil profesional a partir del portal *Glassdoor* [51, 52, 53, 54, 55, 56, 57]. A cada salario bruto se le añade un 30 % adicional correspondiente a cotizaciones sociales: contingencias comunes, desempleo, formación y Fogasa.

Se asume una jornada laboral de 8 h/día, 20 días laborables por mes y 11 meses de trabajo al año, siendo 220 días laborales al año. La planificación se basa en 80 días efectivos de trabajo entre el **3 de marzo y el 20 de junio de 2025**, distribuidos entre los distintos perfiles según su implicación en cada fase.

Cuadro 3.2: Perfiles y costes de personal (según cotización SS 2025)

Perfil	Bruto/año (€)	+31.4 % SS (€)	Total (€)	€/día	€/h
Jefe de Proyecto	40 500	12 717.00	53 217.00	270.92	33.87
Desarrollador Frontend	40 500	12 717.00	53 217.00	270.92	33.87
Desarrollador Backend	30 000	9 420.00	39 420.00	200.10	25.01
Desarrollador RA	23 000	7 222.00	30 222.00	153.68	19.21
Tester / QA	23 500	7 379.00	30 879.00	156.71	19.59
Modelador 3D	12 000	3 768.00	15 768.00	79.84	9.98
Diseñador gráfico	26 000	8 164.00	34 164.00	173.32	21.67

3.5.2. Asignación temporal por perfil

Se ha asignado a cada perfil el número estimado de jornadas necesarias para cubrir las tareas planificadas según el cronograma de trabajo. A continuación, se presenta la estimación de coste directo de personal en función de esas jornadas.

Cuadro 3.3: Asignación de tareas y costes por perfil

Perfil	Días	€/día	Tareas	Coste (€)
Jefe de Proyecto	50	270.92	Estudio inicial, definición de objetivos, planificación, selección de tecnologías, coordinación general	13,546.00
Desarrollador Frontend	16	270.92	Diseño arquitectura frontend, desarrollo UI, navegación, QR, filtros, mapa	4,334.72
Desarrollador Backend	15	200.10	Creación API Django, modelo de datos, autenticación, sincronización backend	3,001.50
Desarrollador RA	7	153.68	Integración AR.js, activación por proximidad, carga de modelos RA	1,075.76
Tester / QA	7	156.71	Pruebas funcionales, pruebas con usuarios, validación y corrección de errores	1,096.97
Modelador 3D	15	79.84	Diseño y preparación de modelos 3D para RA	1,197.60
Diseñador gráfico	20	173.32	Diseño interfaz, estilo visual, iconos y logotipos	3,466.40

3.5.3. Costes directos de material

Los costes directos de material comprenden los recursos físicos y licencias digitales utilizados durante el desarrollo del proyecto. Para estimar su impacto económico real, se ha calculado la amortización proporcional al tiempo de uso, aplicando la fórmula:

$$\text{Amortización} = \frac{\text{Precio unidad} \times \text{Días de uso}}{\text{Días de vida útil}}$$

Cuadro 3.4: Elementos utilizados y coste amortizado durante el desarrollo (80 días)

Elemento	Precio (€)	Vida útil	Uso (días)	Coste amortizado (€)
Portátil Asus ROG Strix G15	999.00	5 años (1825 días)	80	43.79
iPhone 11	749.00	4 años (1460 días)	80	41.03
Ratón Logitech G203	20.99	5 años (1825 días)	80	0.92
Visual Paradigm (mensual)	30.00	30 días	30	30.00
Microsoft 365 Business Standard	12.50	30 días	30	12.50
Visual Studio Code (profesional)	45.00	365 días	80	9.86
Docker Pro	99.00	365 días	80	21.70
GitHub Copilot	100.00	365 días	80	21.92
Overleaf Premium	140.00	365 días	80	30.68
Total amortizado				212.40 €

Cabe destacar que varias de estas herramientas (como Visual Studio Code, Docker o GitHub) disponen de versiones gratuitas o de acceso académico, pero se ha optado por reflejar el coste real de uso profesional para una estimación más realista.

3.5.4. Costes indirectos

Los costes indirectos se han estimado aplicando un coeficiente adicional del 20 % sobre el total de los costes directos (personal y material). Este porcentaje se considera representativo para cubrir gastos generales como el consumo eléctrico, conexión a internet, mantenimiento del equipo, uso de espacios compartidos y otros recursos no imputables directamente al desarrollo del proyecto, pero imprescindibles para su ejecución.

$$\text{Costes indirectos} = (\text{Costes directos de personal} + \text{Costes directos de material}) \times 0,20$$

$$\text{Costes indirectos} = (27\,718,95 + 212,40) \times 0,20 = 5\,586,27$$

3.5.5. Coste total del proyecto

A partir de la suma de los distintos componentes estimados (personal, material e indirectos), se obtiene el coste total simulado del desarrollo del proyecto en un entorno profesional.

Cuadro 3.5: Resumen de costes del proyecto

Concepto	Coste (€)
Costes directos de personal	27 718,95
Costes directos de material	212,40
Costes indirectos (20 %)	5 586,27
Coste total estimado	33 517,62

Esta estimación permite cuantificar de forma realista los recursos necesarios para ejecutar un proyecto de características similares en un entorno empresarial. Aunque el trabajo haya sido desarrollado por una única persona, se ha optado por realizar una simulación profesional con perfiles especializados para reflejar adecuadamente la complejidad técnica y el valor añadido del producto final.

3.6. Viabilidad del proyecto

3.6.1. Viabilidad económica

El análisis de costes realizado en la sección anterior ha permitido cuantificar los recursos necesarios para el desarrollo de la aplicación *DONA'm MÓN*. El coste total estimado, que incluye los costes directos de personal (27.718,95 €), costes directos de material (212,40 €) y un 20 % adicional en concepto de costes indirectos (5.586,27 €), asciende a **33.517,62 €**.

Dado que el proyecto se ha desarrollado sin fines comerciales, el desarrollo ha sido ejecutado por una única persona, utilizando licencias gratuitas, versiones académicas o software de código abierto. Por tanto, los costes no se han asumido en su totalidad. No obstante, se ha realizado una estimación económica completa con el objetivo de evaluar su viabilidad en un contexto profesional o empresarial real, permitiendo así dimensionar correctamente los recursos necesarios en caso de una futura implementación a escala.

El presupuesto puede considerarse razonable si se contempla una posible financiación pública (subvenciones a proyectos culturales, feministas o tecnológicos), financiación universitaria, patrocinio institucional (ayuntamientos, universidades o fundaciones), o apoyo por parte de entidades interesadas en promover la visibilización de mujeres en entornos urbanos y educativos. Además, el impacto social y educativo del proyecto puede justificar su financiación mediante convocatorias de innovación social o género.

3.6.2. Viabilidad legal

Desde el punto de vista legal, el desarrollo de esta aplicación requiere considerar varias cuestiones relevantes:

- **Protección de datos personales (RGPD):** la aplicación permite el registro de usuarios y almacena información relacionada con su ubicación geográfica y su histórico de visitas. Por ello, se deberá garantizar el cumplimiento del Reglamento General de Protección de Datos (RGPD), asegurando la obtención del consentimiento explícito del usuario, el almacenamiento seguro de datos y la posibilidad de acceder, rectificar o eliminar la información personal.
- **Política de privacidad y condiciones de uso:** será necesario incluir una política clara y accesible que informe al usuario sobre el tratamiento de sus datos, el uso de la geolocalización y la finalidad educativa de la aplicación.
- **Derechos de autor y licencias:** todos los modelos 3D, imágenes, textos y recursos utilizados deben contar con licencia libre (Creative Commons, dominio público) o haber sido creados expresamente para el proyecto. En caso contrario, sería necesario solicitar los permisos adecuados.
- **Política de uso de servicios externos:** la aplicación hace uso de tecnologías como *WebView* para integrar experiencias de realidad aumentada desarrolladas con A-Frame y AR.js, por lo que se han respetado sus licencias de uso (ambas de código abierto bajo licencia MIT), garantizando la legalidad del uso de estas herramientas dentro del proyecto.

No se han identificado restricciones legales insalvables que impidan la realización o

publicación de la aplicación. En todo caso, se recomienda aplicar buenas prácticas en el desarrollo seguro, accesible y ético del software.

3.7. Análisis de riesgos

Durante el desarrollo del proyecto *DONA'm MÓN* se han identificado diversos riesgos que pueden afectar negativamente a su cumplimiento en tiempo, coste o calidad. Para su evaluación se ha utilizado un enfoque cualitativo que combina la estimación de la probabilidad de ocurrencia y el impacto en el proyecto, permitiendo establecer niveles de riesgo y su clasificación prioritaria.

3.7.1. Principales riesgos identificados

En la Tabla 3.6 se recogen los principales riesgos detectados, su probabilidad estimada, el impacto medido en días de retraso potencial, su nivel de riesgo ($NR = \text{Probabilidad} \times \text{Impacto}$), la clasificación resultante y la fase del proyecto en la que podrían manifestarse.

Cuadro 3.6: Principales riesgos identificados

Riesgo	Prob. (%)	Impacto (días)	NR	Clasificación	Fase
Retrasos en la integración de la RA con React Native	40	8	3.20	Inaceptable	Ejecución
Cambios en dependencias externas (Mapas, RA, etc.)	30	5	1.50	Alto	Ejecución
Falta de experiencia previa con tecnologías clave	35	6	2.10	Alto	Planificación/Ejecución
Definición ambigua del alcance inicial	25	7	1.75	Alto	Inicio
Estimaciones optimistas en tareas críticas	20	6	1.20	Moderado	Planificación
Problemas legales (datos, localización, RGPD)	15	5	0.75	Moderado	Inicio
Baja disponibilidad de recursos y tiempo personal	30	6	1.80	Alto	Ejecución
Errores funcionales en fases finales	10	5	0.50	Bajo	Cierre

A continuación se describen los riesgos con mayor nivel de prioridad:

1. **Definición ambigua del alcance del proyecto:** Un alcance mal delimitado puede derivar en objetivos poco claros y decisiones incorrectas en fases posteriores del desarrollo.

Probabilidad: Media — Impacto: Alto — Nivel de riesgo: Inaceptable

2. **Estimaciones temporales y de costes imprecisas:** Una planificación optimista puede provocar desviaciones significativas respecto al cronograma o sobrecostes imprevistos.

Probabilidad: Alta — Impacto: Alto — Nivel de riesgo: Inaceptable

3. **Falta de experiencia con tecnologías específicas:** La falta de familiaridad con herramientas clave como AR.js o A-Frame podría ralentizar la implementación de la funcionalidad de realidad aumentada.

Probabilidad: Media — Impacto: Alto — Nivel de riesgo: Inaceptable

- 4. Retrasos en la ejecución de tareas clave:** La elevada carga de trabajo asumida por una sola persona incrementa la probabilidad de bloqueos o demoras.

Probabilidad: Alta — Impacto: Medio — Nivel de riesgo: Alto

- 5. Dependencia de recursos externos:** El uso de librerías de terceros (como AR.js, Mapbox, A-Frame) puede verse afectado por cambios inesperados o falta de mantenimiento.

Probabilidad: Media — Impacto: Medio — Nivel de riesgo: Medio

- 6. Problemas de compatibilidad entre componentes:** La integración entre tecnologías heterogéneas (React Native, Django, AR.js) podría causar errores o conflictos difíciles de prever.

Probabilidad: Baja — Impacto: Alto — Nivel de riesgo: Alto

- 7. Riesgos legales:** A pesar de que no se almacena información sensible, el tratamiento de datos de geolocalización y el uso de usuarios registrados exige el cumplimiento de la normativa vigente en materia de protección de datos (RGPD).

Probabilidad: Baja — Impacto: Medio — Nivel de riesgo: Medio

3.7.2. Matriz de riesgos

La figura 3.3 muestra la matriz de evaluación de riesgos utilizada, que relaciona la probabilidad de ocurrencia con la magnitud del impacto. Esta herramienta facilita la clasificación de los riesgos y la priorización de acciones preventivas.

		IMPACTO EN EL PROYECTO		
		BAJO	MEDIO	ALTO
PROBABILIDAD DE QUE OCURRA (%)	ALTA [70 - 100]	RIESGO MEDIO	RIESGO ALTO	RIESGO INACEPTABLE
	MEDIA [35 - 70]	RIESGO BAJO	RIESGO ALTO	RIESGO INACEPTABLE
	BAJA [10 - 35]	RIESGO BAJO	RIESGO MEDIO	RIESGO ALTO

Figura 3.3: Matriz de priorización de riesgos utilizada

3.7.3. Resumen de evaluación de riesgos

Cuadro 3.7: Resumen cualitativo de evaluación de riesgos

Riesgo	Probabilidad	Impacto	Clasificación
Definición ambigua del alcance	Media	Alto	Inaceptable
Estimaciones temporales imprecisas	Alta	Alto	Inaceptable
Falta de experiencia con RA	Media	Alto	Inaceptable
Retrasos en tareas clave	Alta	Medio	Alto
Dependencia de recursos externos	Media	Medio	Medio
Problemas de compatibilidad tecnológica	Baja	Alto	Alto
Riesgos legales (RGPD)	Baja	Medio	Medio

3.7.4. Estrategias de mitigación

Para minimizar el impacto de los riesgos identificados en el proyecto, se han definido una serie de medidas preventivas orientadas a reducir su probabilidad de ocurrencia o su impacto en caso de materializarse. Estas estrategias se alinean con las buenas prácticas en la gestión de proyectos tecnológicos y tienen carácter proactivo.

- **Definición ambigua del alcance del proyecto:** Se ha trabajado desde el inicio con una delimitación clara del alcance y los objetivos, apoyada por una estructura modular de tareas y entregables. La revisión periódica del cronograma y del backlog funcional permite detectar desviaciones tempranas y corregir posibles ambigüedades.
- **Estimaciones temporales y de costes imprecisas:** La planificación se ha fundamentado en la técnica de estimación por tres valores (optimista, más probable y pesimista), reduciendo la subjetividad y permitiendo incorporar márgenes de seguridad en tareas críticas. Además, se han utilizado referencias de proyectos similares como guía de validación.
- **Falta de experiencia con tecnologías específicas:** Durante la fase de análisis técnico se han realizado pruebas de concepto con las tecnologías clave (AR.js, A-Frame, integración con React Native) para identificar dificultades de forma anticipada. Asimismo, se ha recurrido a documentación oficial, foros especializados y comunidades activas.
- **Retrasos en la ejecución de tareas clave:** Dado que el proyecto es desarrollado por una única persona, se han asignado holguras temporales en tareas críticas y puntos de control semanales para verificar el avance real. El uso de herramientas como Microsoft Project ha permitido detectar desviaciones y replanificar en tiempo real.
- **Dependencia de recursos externos:** Se ha priorizado el uso de herramientas open source con comunidades activas y mantenidas (por ejemplo, AR.js, OpenStreetMap). Además, se han definido posibles alternativas ante fallos o incompatibilidades en recursos externos.

- **Problemas de compatibilidad entre componentes:** La arquitectura del sistema se ha diseñado de forma desacoplada, favoreciendo la integración por medio de interfaces RESTful y estándares abiertos. Esto reduce la posibilidad de conflictos entre tecnologías heterogéneas.
- **Riesgos legales (RGPD):** Aunque no se almacenan datos sensibles, se garantiza el cumplimiento normativo mediante el uso de comunicaciones seguras (HTTPS), el consentimiento del usuario para acceder a la ubicación y la ausencia de almacenamiento persistente de información personal.

3.7.5. Planes de contingencia

Se han previsto planes de contingencia específicos para los riesgos más relevantes. Por ejemplo, si se detectan incompatibilidades graves entre tecnologías, se contemplaría el uso de una solución alternativa basada únicamente en mapas interactivos sin RA, manteniendo la funcionalidad básica de exploración de contenidos educativos geolocalizados.

Capítulo 4

Análisis

Capítulo 5

Diseño

Capítulo 6

Implementación y pruebas

6.1. Implementación

6.2. Pruebas funcionales

6.3. Pruebas de rendimiento

6.4. Pruebas de usabilidad

Capítulo 7

Conclusiones

7.1. Revisión de costes

7.2. Conclusiones

7.3. Trabajo futuro

Hablar con Por Ti Mujer y la Asamblea de Mujeres de Valencia por si les interesa que la aplicación sea un recurso para sus rutas. Poder descargar los datos de la base de datos en formato CSV o JSON para su uso en otros proyectos. Poder añadir nuevos puntos de interés a la base de datos desde la aplicación móvil. Poder añadir nuevos puntos de interés a la base de datos desde el panel de administración. Página de contacto para que los usuarios puedan enviar sugerencias o comentarios sobre la aplicación. Tener un apartado de preguntas frecuentes (FAQ) para resolver dudas comunes de los usuarios. Tener una sección de noticias o novedades sobre la aplicación y las mujeres que aparecen en ella. Tener una página para administrar todo lo relaciona con la aplicación. Expandir la aplicación a otras ciudades de España o del mundo.

Apéndice A

Apéndice

A.1. Ejemplos del lenguaje de marcado Latex

Antes de empezar a escribir la memoria debes leerte, al menos, los dos primeros capítulos del clásico de Tobias Oetker [?]¹.

El Apéndice A.1 recoge algunos ejemplos de edición con LATEX.

This document is an example of BibTeX using in bibliography management. Three items are cited: *The LATEX Companion* book [?], the Einstein journal paper [?], and the Donald Knuth's website [?]. The LATEX related items are [?, ?]².

Texto en el párrafo 1.

Texto en el párrafo 2.

Texto en el párrafo 3.

■ Consideración 1

■ Consideración 2

1. Punto 1

2. Punto 2

A continuación se muestra una ecuación:

$$\int_0^1 \frac{1}{x^2 + 1} dx$$

Podemos incluir imágenes en formato: png, pdf o jpg.

En la figura A.1 se muestra un diagrama realizado con <https://www.yworks.com/products/yed>:

¹La última versión puede encontrarse en <http://tobi.oetiker.ch/lshort/lshort.pdf>.

²Esto está tomado de https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex

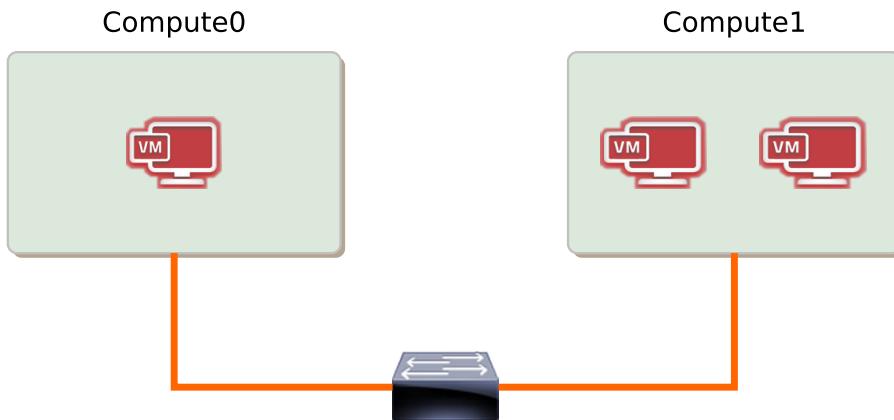


Figura A.1: Esta es una figura que latex decide donde colocar (floating) en el documento.

Imagen 1

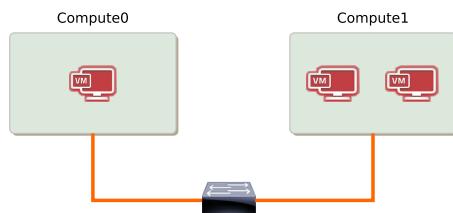
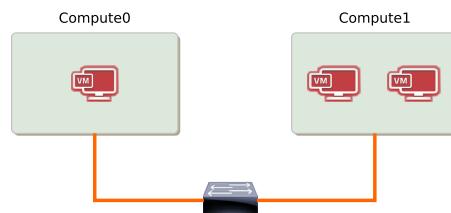


Imagen 2



Este es un ejemplo de una tabla:

Columna 1	Columna 2
1	2

O la misma tabla centrada:

Columna 1	Columna 2
1	2

Para generar el fichero PDF:

```
pdflatex ejemplo-memoria.tex
bibtex ejemplo-memoria
pdflatex ejemplo-memoria.tex
```

También se puede usar `latexmk` que automáticamente regenera la bibliografía.

```
latexmk -pdf ejemplo-memoria.tex
```

Bibliografía

- [1] UNESCO. Science report: The race against time for smarter development. <https://unesdoc.unesco.org/ark:/48223/pf0000377433>, 2021. United Nations Educational, Scientific and Cultural Organization.
- [2] Ministerio de Ciencia e Innovación. Científicas en cifras 2021: Estadísticas e indicadores de la (des)igualdad de género en la formación y profesión científica. https://www.ciencia.gob.es/dam/jcr:dc8689c4-2c47-4aaf-97ce-874bd0b5a081/Cientificas_en_Cifras_2021.pdf, 2022. Secretaría General de Investigación.
- [3] Jon Agar. Constant touch: A global history of the mobile phone, 2004. ISBN 978-1840465413.
- [4] Joel West and Jason Dedrick. Globalization and the evolution of the mobile phone industry. *International Journal of Technology and Globalisation*, 5(1/2):65–80, 2010.
- [5] Jamie Murphy. Mobile marketing: The future of marketing, or just another fad? *International Journal of Mobile Marketing*, 9(1):47–59, 2014.
- [6] StatCounter. Mobile operating system market share worldwide. <https://gs.statcounter.com/os-market-share/mobile/worldwide>, 2024.
- [7] Statista. Number of smartphone users worldwide from 2016 to 2024. <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>, 2024.
- [8] AndroidPolice. Android versus ios software updates revisited, 2017.
- [9] Gartner. Market guide for multiexperience development platforms, 2022.
- [10] Data.ai. State of mobile 2023, 2023.
- [11] App Annie. State of mobile report 2022, 2022.
- [12] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997. <https://doi.org/10.1162/pres.1997.6.4.355>.
- [13] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Foundations and Trends in Human–Computer Interaction*, 8(2–3):73–272, 2015. <https://doi.org/10.1561/1100000049>.
- [14] Steven Feiner, Blair MacIntyre, and Dorée Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, 1993. <https://doi.org/10.1145/159544.159587>.

- [15] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94, 1999. <https://doi.org/10.1109/IWAR.1999.803809>.
- [16] Daniel Wagner and Dieter Schmalstieg. First steps towards handheld augmented reality. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, pages 127–135, 2003. <https://doi.org/10.1109/ISWC.2003.1241409>.
- [17] Apple. Arkit: Augmented reality for ios. <https://developer.apple.com/augmented-reality/>, 2017.
- [18] Google. Arcore: Google developers. <https://developers.google.com/ar>, 2018.
- [19] J. Paavilainen, J. Hamari, J. Stenros, and J. Kinnunen. The pokémon go experience: A location-based augmented reality mobile game goes mainstream. In *Proceedings of CHI 2017*, 2017.
- [20] Microsoft. Hololens product page. <https://www.microsoft.com/en-us/hololens>, 2016.
- [21] D. W. F. van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *The International Journal of Virtual Reality*, 9(2):1–20, 2010.
- [22] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR 2007*, 2007.
- [23] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 234–241, 1997. <https://doi.org/10.1145/258549.258715>.
- [24] M. Billinghurst and H. Kato. Collaborative mixed reality. In *ISMAR 2002*, 2002.
- [25] Apple. Introducing apple vision pro. <https://www.apple.com/apple-vision-pro/>, 2024.
- [26] Women's legacy. proyecto europeo cofinanciado por la unión europea, 2022. Conselleria d'Educació, Cultura i Esport de la Generalitat Valenciana.
- [27] Dones de ciència: Murals que inspiren, 2020. Las Naves & UPV.
- [28] Callejero con perspectiva de género: iniciativas y actuaciones, 2023. Ayuntamiento de València.
- [29] Asociación Por Ti Mujer. Ruta violeta de mujeres en valencia, 2023. Recuperado de <https://asociacionportimujer.org/ruta-violeta-de-mujeres-en-valencia>.
- [30] Assemblea Feminista de València. Ruta pels espais del patronato de protección a la mujer, 2023. Recuperado de <https://www.feministas.org/ruta-pels-espais-del-patronato-de.html>.
- [31] Assemblea Feminista de València. Ruta de memorias lesbianas en valencia: visibilización y lucha feminista, 2024. Recuperado de <https://www.levante-emv.com/valencia/2024/10/27/mapa-memorias-lesbianas-valencia-lesbianismo-lgtbi-ruta-assemblea-feminista-feminismo-110410679.html>.
- [32] R. Guijarro-Garzón et al. Pioneras de la medicina: Concepción aleixandre y la visibilización de la mujer en la ciencia. *Revista de Historia de la Medicina*, 12(2):134–148, 2021.

- [33] Carmen Alborch. *Solas*. Espasa Calpe, 1999.
- [34] Amelia González. *Clara Campoamor: el sufragio femenino en España*. Cátedra, 2006.
- [35] Carmen Rueda Ramos. *Isabel de Villena y la espiritualidad femenina en la Valencia del siglo XV*. Tirant lo Blanch, 2013.
- [36] J. Martínez Pérez. *Médicas en Valencia: mujeres que abrieron camino*. Editorial UV, 2019.
- [37] Vandana Shiva. *Earth Democracy: Justice, Sustainability, and Peace*. South End Press, 2005.
- [38] Google Developers. Google maps platform documentation. <https://developers.google.com/maps/documentation>.
- [39] Mapbox. Mapbox documentation. <https://docs.mapbox.com/>.
- [40] OpenStreetMap Foundation. Openstreetmap wiki. <https://wiki.openstreetmap.org/>.
- [41] React Native Maps Contributors. react-native-maps: React native mapview component for ios + android. <https://github.com/react-native-maps/react-native-maps>.
- [42] Django Software Foundation. Django documentation, 2024.
- [43] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of database systems*. Pearson, 7th edition, 2017.
- [44] Michael Stonebraker and Rick Cattell. 10 rules for scalable performance in ‘simple operation’ datastores. *Communications of the ACM*, 54(6):72–80, 2011.
- [45] Antonio Mele Vincent. *Mastering Django: Core*. Independently published, 2020.
- [46] Oracle Corporation. Mysql 8.0 reference manual, 2023.
- [47] MariaDB Foundation. Mariadb vs mysql, 2023.
- [48] Inc. MongoDB. Mongodb manual, 2024.
- [49] Django Software Foundation. Using sqlite with django, 2024.
- [50] Apache Software Foundation. Apache cassandra documentation, 2023.
- [51] Glassdoor. Sueldo de jefe de proyecto. https://www.glassdoor.es/Sueldos/jefe-de-proyectos-sueldo-SRCH_KO0,17.htm, 2025. Consultado el 11 de junio de 2025.
- [52] Glassdoor. Sueldo de desarrollador react native. https://www.glassdoor.es/Sueldos/react-native-developer-sueldo-SRCH_KO0,22.htm, 2025. Consultado el 11 de junio de 2025.
- [53] Glassdoor. Sueldo de desarrollador python/django. https://www.glassdoor.es/Sueldos/django-developer-sueldo-SRCH_KO0,16.htm, 2025. Consultado el 11 de junio de 2025.
- [54] Glassdoor. Sueldo de qa tester. https://www.glassdoor.es/Sueldos/qa-tester-sueldo-SRCH_KO0,9.htm, 2025. Consultado el 11 de junio de 2025.
- [55] Glassdoor. Sueldo de modelador 3d. https://www.glassdoor.es/Sueldos/modelador-de-3d-sueldo-SRCH_KO0,15.htm, 2025. Consultado el 11 de junio de 2025.

- [56] Glassdoor. Sueldo de diseñador gráfico. https://www.glassdoor.es/Sueldos/disenador-grafico-sueldo-SRCH_KO0,17.htm, 2025. Consultado el 11 de junio de 2025.
- [57] Glassdoor. Sueldo de desarrollador de realidad aumentada/virtual. https://www.glassdoor.es/Sueldos/virtual-reality-developer-sueldo-SRCH_KO0,25.htm, 2025. Consultado el 13 de junio de 2025.