

Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática de Albacete

Escuela Superior de Informática de Ciudad Real

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial

Diseño y Desarrollo de un Asistente Virtual basado en LLMs para Aplicaciones de Realidad Mixta en Atención Primaria

José Manuel Bastante Rodríguez

Octubre, 2026

Escuela Superior de Ingeniería Informática Albacete

Escuela Superior de Informática

TRABAJO FIN DE MÁSTER

Máster Universitario en Inteligencia Artificial

Diseño y Desarrollo de un Asistente Virtual basado en LLMs para Aplicaciones de Realidad Mixta en Atención Primaria

Autor: José Manuel Bastante Rodríguez

Director: Santiago Sánchez Sobrino

Codirector: David Vallejo Fernández

UCLM Universidad Castilla-La Mancha

Octubre, 2026

A mi hermana y a mis padres. Por confiar siempre en mí, por el cariño y por el apoyo que me habéis dado.

Declaración de autoría

Yo, José Manuel Bastante Rodríguez , con DNI 05736476T , declaro que soy el único autor del trabajo fin de máster titulado “Diseño y Desarrollo de un Asistente Virtual basado en LLMs para Aplicaciones de Realidad Mixta en Atención Primaria ”, que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual, y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Ciudad Real, a 7 de octubre de 2026

Fdo.: José Manuel Bastante Rodríguez

Resumen

El presente Trabajo Fin de Máster aborda el diseño e implementación de un asistente virtual inteligente basado en Grandes Modelos de Lenguaje para su integración en aplicaciones de realidad mixta en el ámbito de la atención primaria, en el marco del proyecto Plataforma de Realidad Mixta para 5G.

El sistema desarrollado combina la técnica de Generación Aumentada por Recuperación con un modelo de lenguaje alojado en la infraestructura de la Universidad de Castilla-La Mancha, manteniendo un mayor control sobre el tratamiento de los datos clínicos. El asistente permite al personal sanitario consultar documentación médica en lenguaje natural, recibiendo respuestas fundamentadas en los documentos indexados y con citas explícitas a las fuentes utilizadas.

Entre las funcionalidades implementadas destacan las siguientes: la gestión de sesiones clínicas con persistencia en base de datos, la extracción automática de tareas a partir de las respuestas del agente, el comportamiento proactivo mediante resúmenes y preguntas sugeridas al cargar nuevos documentos, la entrada por voz mediante transcripción automática con Whisper, y una interfaz web desarrollada con Angular 22 que integra todas estas capacidades en un entorno intuitivo y accesible.

El sistema ha sido evaluado mediante pruebas técnicas y funcionales utilizando como referencia la guía ESC 2024 sobre el manejo de la presión arterial elevada y la hipertensión. Los resultados obtenidos permiten caracterizar el comportamiento del mecanismo de recuperación documental, la generación de respuestas, la trazabilidad de las fuentes y la respuesta del sistema ante consultas para las que no existe evidencia suficiente en la documentación disponible.

Abstract

This Master's Thesis addresses the design and implementation of an intelligent virtual assistant based on Large Language Models for its integration into mixed reality applications in the field of primary care, within the framework of the Mixed Reality Platform for 5G project.

The developed system combines Retrieval-Augmented Generation with a language model hosted on the infrastructure of the University of Castilla-La Mancha, providing greater control over the processing of clinical data. The assistant enables healthcare professionals to query medical documentation using natural language and receive answers grounded in the indexed documents, together with explicit references to the sources used.

The main implemented functionalities include the management of clinical sessions with database persistence, automatic extraction of tasks from the assistant's responses, proactive behaviour through summaries and suggested questions when new documents are uploaded, voice input through automatic transcription with Whisper, and a web interface developed with Angular 22 that integrates all these capabilities into an intuitive and accessible environment.

The system has been evaluated through technical and functional tests using the 2024 ESC Guidelines for the management of elevated blood pressure and hypertension as the reference document. The obtained results make it possible to characterise the behaviour of the document retrieval mechanism, response generation, source traceability, and the system's behaviour when queries cannot be supported by sufficient evidence in the available documentation.

Agradecimientos

Tras finalizar este Trabajo de Fin de Máster, doy por finalizada una nueva etapa de mi vida, tan exigente como enriquecedora, en la que he crecido tanto personal como profesionalmente.

En primer lugar, quiero agradecer a mi familia, en especial a mis padres José Vicente y Blanca y a mi hermana Elena, por su apoyo incondicional, por confiar siempre en mí y por estar ahí en los momentos más difíciles.

A Óscar, amigo y compañero de máster, por haber recorrido juntos esta etapa. Su apoyo y compañía han hecho todo mucho más llevadero.

A mis tutores Santiago Sánchez Sobrino y David Vallejo Fernández, por su orientación, su disponibilidad y su implicación a lo largo del desarrollo de este trabajo.

A la Escuela Superior de Informática de la UCLM por haber facilitado el marco académico para la realización de este trabajo.

Y, finalmente, agradecerme a mí mismo por seguir adelante y no tirar la toalla. Sin mi esfuerzo y disciplina, no habría sido posible.

En definitiva, gracias a todos los que de una u otra forma habéis formado parte de esta etapa.

Acrónimos

TFM	Trabajo Fin de Máster
TFG	Trabajo Fin de Grado
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
IA	Inteligencia Artificial
API	Application Programming Interface
UCLM	Universidad de Castilla-La Mancha
MRP-5G	Mixed Reality Platform for 5G
MR	Realidad Mixta
RGPD	Reglamento General de Protección de Datos
AI Act	Reglamento de Inteligencia Artificial de la UE
AMIE	Articulate Medical Intelligence Explorer
MRR	Mean Reciprocal Rank

Índice general

1 Introducción **1**

1.1 Motivación 1

1.2 Descripción del problema 3

1.3 Solución propuesta 4

1.4 Estructura del documento 6

2 Objetivos **7**

2.1 Objetivo general. 7

2.2 Objetivos específicos 7

3 Contexto tecnológico y estado del arte **9**

3.1 Grandes modelos de lenguaje en el ámbito sanitario 9

3.2 Generación aumentada por recuperación 10

3.3 Realidad Mixta aplicada a salud y telemedicina 11

3.4 Asistentes clínicos basados en Inteligencia Artificial 11

3.5 Marco regulatorio 12

3.6 El proyecto MRP-5G 13

3.7 Trabajo previo en MRP-5G 14

3.8 Posicionamiento del presente TFM 16

4 Metodología 17

4.1 Enfoque metodológico 17

4.2 Marco tecnológico 18

4.3 Medios hardware 19

4.4 Planificación temporal 20

4.5 Organización del código fuente 20

5 Arquitectura 21

5.1 Visión general de la arquitectura del sistema 21

5.2 Backend y persistencia 23

5.2.1 Organización del backend 24

5.2.2 Modelo basado en sesiones 24

5.2.3 Persistencia relacional y vectorial 26

5.2.4 Servicios compartidos. 26

5.3 Arquitectura de recuperación y generación 28

5.3.1 Procesamiento documental 28

5.3.2 Representación y recuperación semántica. 28

5.3.3 Diseño del pipeline RAG. 29

5.3.4 Contexto conversacional y trazabilidad 30

5.3.5 Generación progresiva y funcionalidades proactivas 30

5.4 Frontend web 31

5.5 Frontend de realidad mixta 32

5.5.1 Separación de responsabilidades 33

5.5.2 Comunicación con el backend 33

5.5.3 Interacción y paneles espaciales 34

5.5.4 Protocolo clínico 34

5.6 Configuración y despliegue 36

6 Resultados 37

6.1 Resultados funcionales del prototipo 37

6.2 Metodología de evaluación 40

6.3 Evaluaciones realizadas 42

6.3.1 Recuperación de información 42

6.3.2 Respuestas generadas 43

6.3.3 Comportamiento ante ausencia de evidencia 44

6.3.4 Rendimiento temporal 45

6.4 Análisis de costes 46

7 Conclusiones y trabajo futuro 47

7.1 Cumplimiento de los objetivos 47

7.2 Conclusiones generales 48

7.3 Competencias adquiridas 49

7.4 Líneas de trabajo futuro 50

Referencias bibliográficas 51

A Guía de instalación y ejecución 55

A.1 Repositorio del proyecto 55

A.2 Requisitos previos 56

A.3 Configuración del entorno 57

A.4 Despliegue mediante Docker Compose 58

A.5 Ejecución de la aplicación de realidad mixta 59

B Preguntas utilizadas en la evaluación 61

B.1 Preguntas con evidencia documental 61

B.2 Preguntas sin evidencia documental. 63

Índice de figuras

3.1 Interfaces y escenarios de uso de la plataforma Mixed Reality Platform for 5G (MRP-5G) 15

4.1 Tecnologías y herramientas empleadas 19
4.2 Planificación temporal del proyecto 20

5.1 Arquitectura general del sistema 22
5.2 Organización interna del backend 23
5.3 Modelo de datos basado en sesiones 25
5.4 Pipeline RAG 29
5.5 Arquitectura de la aplicación de realidad mixta 32
5.6 Arquitectura de despliegue 36

6.1 Resultados funcionales de la aplicación web 38
6.2 Interacción con el asistente mediante RAG 39
6.3 Vista general de la aplicación. 39
6.4 Funcionamiento del prototipo de realidad mixta en Meta Quest 3. 40

Índice de tablas

6.1 Comparación de los modelos de embeddings evaluados. 42
6.2 Resultados de la evaluación de la recuperación documental. 43
6.3 Resultados de la evaluación extremo a extremo. 44
6.4 Resumen de los tiempos obtenidos durante la evaluación. 45
6.5 Estimación de los costes asociados al desarrollo del proyecto. 46

B.1 Composición del conjunto de preguntas utilizado en la evaluación. . . . 61
B.2 Preguntas de evaluación con evidencia documental. 62
B.3 Preguntas de evaluación sin evidencia documental. 64

1. Introducción

1.1. Motivación

La atención primaria constituye el primer nivel de contacto entre la población y el sistema sanitario y desempeña un papel fundamental en la prevención, diagnóstico, tratamiento y seguimiento de un amplio conjunto de patologías [WHO and UNICEF, 2020]. Los profesionales que trabajan en este ámbito deben tomar decisiones de forma continua, consultar protocolos clínicos, interpretar información procedente de distintas fuentes y mantener un seguimiento adecuado de cada paciente, todo ello dentro de un entorno caracterizado por una elevada carga asistencial y una disponibilidad de tiempo limitada [Irving et al., 2017].

En este contexto, las tecnologías basadas en Inteligencia Artificial (IA) ofrecen nuevas posibilidades para facilitar el acceso a información clínica relevante y proporcionar herramientas de apoyo durante la práctica asistencial [Topol, 2019]. En particular, los Large Language Models (LLMs) permiten interactuar con sistemas informáticos mediante lenguaje natural, facilitando la realización de consultas complejas sin necesidad de utilizar mecanismos tradicionales de búsqueda [Brown et al., 2020, Singhal et al., 2023].

Sin embargo, su utilización directa en el ámbito sanitario presenta importantes limitaciones [Bedi et al., 2025, Singhal et al., 2023]. Estos modelos pueden generar información incorrecta con apariencia de veracidad, utilizar conocimiento desactualizado o no disponer de acceso a la documentación específica utilizada por un determinado profesional o centro sanitario.

Una de las técnicas empleadas para reducir estas limitaciones es la generación aumentada mediante recuperación, conocida como Retrieval-Augmented Generation (RAG) [Lewis et al., 2020, Amugongo et al., 2025]. Este enfoque combina un modelo generativo con un mecanismo de recuperación de información, de forma que la respuesta se construye utilizando como contexto fragmentos obtenidos de una base documental previamente indexada. De esta manera, resulta posible orientar las respuestas hacia documentación concreta y proporcionar al usuario información sobre las fuentes utilizadas durante la generación.

Paralelamente, la Realidad Mixta (MR) permite integrar información digital dentro del entorno físico del usuario [Milgram and Kishino, 1994]. En el ámbito sanitario, esta característica resulta especialmente interesante, ya que permite presentar información al profesional sin necesidad de recurrir continuamente a una pantalla convencional y manteniendo una mayor libertad de movimiento. La combinación de interfaces de MR, interacción mediante voz y sistemas basados en IA plantea, por tanto, una oportunidad para desarrollar herramientas de apoyo integradas en el flujo de trabajo del profesional [Worlikar et al., 2023, García et al., 2025a].

El presente Trabajo Fin de Máster (TFM) se enmarca dentro del proyecto MRP-5G, una iniciativa de investigación de la Universidad de Castilla-La Mancha (UCLM) orientada al desarrollo de una plataforma de MR para proporcionar soporte remoto en escenarios de atención primaria [García et al., 2025a].

MRP-5G plantea la combinación de tecnologías de MR, comunicaciones de baja latencia e IA para facilitar la colaboración entre profesionales sanitarios. La plataforma contempla distintos escenarios de utilización, como centros de atención primaria, atención domiciliaria y situaciones de emergencia, en los que un profesional puede utilizar un dispositivo de MR para acceder a información digital y recibir apoyo durante la atención al paciente.

Dentro de esta arquitectura global, una de las líneas de trabajo consiste en proporcionar mecanismos de asistencia inteligente que permitan al profesional acceder a información clínica de forma rápida y contextualizada. El presente TFM se sitúa en esta línea y aborda específicamente el diseño de un asistente virtual basado en modelos de lenguaje, recuperación documental e interacción mediante MR.

1.2. Descripción del problema

La aplicación de modelos de lenguaje en entornos sanitarios plantea varios desafíos que deben ser considerados antes de su utilización como herramienta de apoyo [Bedi et al., 2025].

Uno de estos desafíos está relacionado con la privacidad y la protección de los datos utilizados durante la interacción con el sistema. La información clínica puede contener datos personales especialmente sensibles y los datos relativos a la salud están considerados como categorías especiales de datos personales por el Reglamento General de Protección de Datos [European Union, 2016]. Por este motivo, resulta conveniente limitar la exposición de esta información a servicios externos y mantener un mayor control sobre la infraestructura encargada de procesarla. En este trabajo, esta consideración se refleja en la utilización de un modelo de lenguaje ejecutado en infraestructura controlada por la UCLM, evitando que las consultas deban enviarse a una Application Programming Interface (API) comercial externa para realizar la inferencia.

Otro desafío es la fiabilidad de la información generada. Los LLMs pueden producir respuestas incorrectas o no fundamentadas, fenómeno habitualmente denominado alucinación [Ji et al., 2023]. En un contexto sanitario, este comportamiento resulta especialmente problemático debido a que la información proporcionada puede influir en la toma de decisiones de un profesional. Además, el conocimiento interno de estos modelos está limitado por los datos empleados durante su entrenamiento y no incluye necesariamente protocolos clínicos específicos, documentos internos o información incorporada posteriormente.

Otro aspecto relevante es que el acceso a la información debe realizarse de forma que el profesional pueda identificar el origen de los datos utilizados en una respuesta. No resulta suficiente obtener una contestación generada en lenguaje natural; es necesario facilitar mecanismos que permitan consultar la documentación asociada y mantener una trazabilidad básica entre la información recuperada y la respuesta proporcionada.

Un desafío adicional es la integración de este tipo de asistente dentro del flujo de trabajo clínico. Una interfaz tradicional basada exclusivamente en teclado y pantalla puede resultar poco adecuada en determinadas situaciones. La interacción mediante voz y el empleo de dispositivos de MR permiten plantear formas alternativas de consulta, pero introducen nuevos requisitos relacionados con la disposición espacial de la información, la legibilidad, la interacción mediante controladores y la adaptación de las respuestas a un espacio de visualización limitado [Billinghurst et al., 2015, Worlikar et al., 2023].

Asimismo, los protocolos clínicos suelen estar formados por secuencias de decisiones dependientes del estado del paciente [Peleg, 2013]. Representar estos procedimientos en una interfaz de MR requiere proporcionar al usuario una visión clara del camino seguido, las opciones disponibles y los posibles pasos posteriores, evitando al mismo tiempo sobrecargar visualmente la escena.

Finalmente, un asistente clínico resulta más útil si es capaz de mantener contexto entre distintas interacciones y ayudar al usuario a realizar un seguimiento de la información generada durante una sesión. Esto requiere mecanismos para organizar conversaciones, documentos, tareas y preferencias asociadas al usuario, además de una arquitectura que permita acceder a estas funcionalidades desde diferentes interfaces.

1.3. Solución propuesta

Para responder a las necesidades identificadas se ha diseñado una arquitectura modular en la que las distintas formas de interacción con el usuario comparten un núcleo común de servicios, persistencia y acceso a la información. De esta forma, las capacidades del asistente no quedan ligadas a una interfaz concreta y pueden reutilizarse desde diferentes clientes.

La arquitectura se organiza conceptualmente en cuatro bloques principales. En primer lugar, se encuentran las interfaces de interacción con el usuario. En segundo lugar, una capa central de servicios coordina la lógica de la aplicación y las operaciones realizadas por los distintos clientes. Un tercer bloque se encarga de la persistencia de la información y de la recuperación documental necesaria para contextualizar las consultas. Finalmente, un componente de generación utiliza la información recuperada para construir las respuestas del asistente mediante un LLM.

Esta separación permite centralizar la lógica del sistema, mantener un contexto común entre las distintas interfaces y evitar la duplicación de funcionalidades. Asimismo, desacopla la interacción con el usuario de los mecanismos de persistencia, recuperación y generación, facilitando la evolución independiente de cada componente.

Sobre esta arquitectura se han implementado dos interfaces principales de acceso. La primera es una aplicación web, concebida como entorno general de gestión del sistema, desde la que el usuario puede autenticarse, crear y recuperar sesiones, incorporar documentación, realizar consultas al asistente, consultar las fuentes utilizadas, gestionar tareas y utilizar las funcionalidades de simulación de casos clínicos.

La segunda interfaz es una aplicación de MR orientada a Meta Quest 3. Esta permite acceder al asistente mediante voz, visualizar las respuestas dentro del entorno físico del usuario mediante *passthrough* e interactuar con protocolos clínicos representados mediante un árbol de decisión progresivo. Los paneles virtuales pueden además manipularse espacialmente para adaptar su posición y orientación a las necesidades del usuario.

El núcleo común del sistema se encarga de gestionar usuarios, sesiones, conversaciones, documentos y tareas, así como de coordinar la recuperación semántica de información, la transcripción de voz y la comunicación con el modelo de lenguaje. Para ello, la solución utiliza persistencia estructurada para la información asociada a las sesiones y almacenamiento vectorial para las representaciones de los fragmentos documentales.

El mecanismo de consulta se basa en una arquitectura RAG [Lewis et al., 2020]. Los documentos incorporados al sistema se procesan, se dividen en fragmentos y se representan mediante *embeddings* antes de ser indexados. Cuando el usuario formula una consulta, se recuperan los fragmentos considerados más relevantes dentro del contexto de la sesión activa y se incorporan a la información proporcionada al modelo de lenguaje. El sistema conserva además los metadatos de procedencia de dichos fragmentos, permitiendo mostrar posteriormente las fuentes asociadas a cada respuesta.

La solución incorpora también mecanismos orientados a proporcionar continuidad entre interacciones. Las conversaciones se organizan mediante sesiones persistentes, los documentos quedan asociados a cada contexto de trabajo y las respuestas pueden dar lugar a la generación de preguntas sugeridas o a la identificación automática de tareas. De esta forma, el asistente no se limita a un esquema aislado de pregunta y respuesta, sino que mantiene información contextual y permite realizar un seguimiento de la actividad desarrollada durante una sesión.

La solución desarrollada tiene carácter experimental y está concebida como herramienta de apoyo. No pretende sustituir el criterio del profesional sanitario ni realizar diagnósticos de forma autónoma, sino explorar la integración de técnicas de IA, recuperación de información y MR en un entorno orientado a atención primaria.

1.4. Estructura del documento

El resto de la memoria se organiza de la siguiente forma:

- *Capítulo 1: Introducción.*

  Se presenta la motivación del trabajo, los principales problemas abordados y una visión general de la solución desarrollada.

- *Capítulo 2: Objetivos.*

  Se define el objetivo general del TFM y los objetivos específicos establecidos para su consecución.

- *Capítulo 3: Contexto tecnológico y estado del arte.*

  Se revisan las principales tecnologías y trabajos relacionados con los LLMs, los sistemas RAG, su aplicación en el ámbito sanitario y el uso de tecnologías de MR como herramienta de apoyo clínico. También se analiza el trabajo previo más próximo al sistema desarrollado.

- *Capítulo 4: Metodología.*

  Se describe la metodología de trabajo seguida durante el desarrollo, la evolución del proyecto, su planificación y las principales tecnologías y recursos empleados.

- *Capítulo 5: Arquitectura.*

  Se presenta la arquitectura de la solución y se describen en detalle sus principales componentes: aplicación web, servicios de *backend*, sistema RAG, gestión de sesiones y tareas, funcionalidades de voz y aplicación de MR.

- *Capítulo 6: Resultados.*

  Se presentan los resultados funcionales del prototipo y los resultados de la evaluación realizada, así como una estimación de los costes asociados al desarrollo.

- *Capítulo 7: Conclusiones y trabajo futuro.*

  Se analiza el grado de cumplimiento de los objetivos planteados, las principales conclusiones, se justifican las competencias abordadas y se recogen las líneas de trabajo futuro.

2. Objetivos

En este capítulo se describen los objetivos planteados al inicio del presente TFM. Estos se estructuran en un objetivo general y una serie de objetivos específicos necesarios para su consecución.

2.1. Objetivo general

El objetivo general de este TFM es diseñar e implementar un asistente virtual contextual y proactivo, basado en LLMs, que proporcione soporte al personal de atención primaria en el marco del proyecto MRP-5G [García et al., 2025a]. El sistema debe permitir consultar documentación clínica mediante lenguaje natural, mantener el contexto de las interacciones, facilitar el seguimiento de información relevante y ofrecer acceso a estas funcionalidades tanto desde una aplicación web como desde una aplicación de MR para Meta Quest 3 integrando al asistente con la navegación de protocolos clínicos.

2.2. Objetivos específicos

Para alcanzar el objetivo general, se han definido los siguientes objetivos específicos:

- **OE1. Estudio de arquitecturas y tecnologías para la integración de LLMs**: analizar arquitecturas, técnicas de recuperación semántica, sistemas RAG y herramientas actuales que permitan integrar modelos de lenguaje en aplicaciones interactivas orientadas al ámbito sanitario.

- **OE2. Diseño del asistente virtual contextual:** diseñar un asistente capaz de responder a consultas formuladas por personal sanitario mediante lenguaje natural, utilizando documentación clínica como contexto, priorizando la claridad, la trazabilidad de las fuentes y la utilidad de las respuestas.
- **OE3. Desarrollo del *backend* del asistente:** implementar un prototipo *backend* encargado de gestionar usuarios, sesiones, conversaciones, documentos y tareas, así como de coordinar la recuperación semántica de información, la comunicación con el modelo de lenguaje y la persistencia de los datos del sistema, teniendo en cuenta aspectos relacionados con el rendimiento y la confidencialidad de los datos.
- **OE4. Desarrollo de mecanismos de interacción y gestión:** proporcionar una interfaz que permita gestionar sesiones, documentos y tareas, consultar al asistente y acceder a las fuentes utilizadas durante la generación de las respuestas, sirviendo además como entorno de apoyo para la prueba y gestión del sistema.
- **OE5. Incorporación de comportamiento contextual y proactivo:** ampliar el comportamiento del asistente más allá de un esquema aislado de pregunta y respuesta mediante mecanismos de persistencia del contexto, generación de preguntas sugeridas, extracción automática de tareas y seguimiento de información relevante.
- **OE6. Definición e integración de mecanismos de interacción multimodal:** incorporar mecanismos de interacción adecuados para su utilización desde una aplicación de MR, incluyendo entrada mediante voz, visualización espacial de la información e interacción con los elementos de la interfaz.
- **OE7. Desarrollo e integración de la aplicación de MR:** desarrollar una aplicación para Meta Quest 3 que permita acceder al asistente dentro del entorno físico del usuario y recorrer protocolos clínicos mediante una representación interactiva en forma de árbol de decisión progresivo.
- **OE8. Evaluación técnica y funcional del prototipo:** validar el funcionamiento del asistente mediante escenarios simulados, evaluando la recuperación de información, la precisión y fundamentación de las respuestas, los tiempos de respuesta y el comportamiento del sistema ante diferentes consultas. Asimismo, evaluar la usabilidad percibida del prototipo mediante la participación de profesionales sanitarios.

3. Contexto tecnológico y estado del arte

En este capítulo se presenta el contexto tecnológico y científico del presente TFM. Se revisan el uso de LLMs en el ámbito sanitario, las arquitecturas RAG, la MR aplicada a salud, algunos asistentes clínicos representativos y el marco regulatorio relevante. Finalmente, se describe el proyecto MRP-5G, el trabajo previo más relacionado y el posicionamiento de la propuesta desarrollada.

3.1. Grandes modelos de lenguaje en el ámbito sanitario

Los grandes modelos de lenguaje o LLMs son modelos de aprendizaje profundo entrenados sobre grandes cantidades de texto para modelar regularidades del lenguaje y generar secuencias a partir de un contexto. Gran parte de los modelos actuales se basan en la arquitectura Transformer [Vaswani et al., 2017], cuya evolución ha permitido abordar tareas como generación de texto, respuesta a preguntas, resumen o interacción conversacional [Brown et al., 2020].

Estas capacidades han impulsado su aplicación en medicina para tareas como respuesta a preguntas clínicas, generación y resumen de documentación, extracción de información, la comunicación con pacientes y el apoyo a profesionales sanitarios. Singhal et al. mostraron que estos modelos pueden adquirir una cantidad considerable de conocimiento clínico, aunque señalaron que su evaluación debe considerar también aspectos como factualidad, riesgo y sesgos [Singhal et al., 2023].

La evidencia en entornos asistenciales reales sigue siendo limitada. Bedi et al. observaron que solo una pequeña proporción de los estudios publicados utilizaba datos procedentes

de atención real a pacientes y que dimensiones como incertidumbre, sesgos o condiciones de despliegue se evaluaban con menor frecuencia [Bedi et al., 2025]. De forma similar, sistemas conversacionales como Articulate Medical Intelligence Explorer (AMIE) han mostrado resultados prometedores en escenarios simulados, aunque todavía requieren validación adicional antes de trasladarse a la práctica clínica [Tu et al., 2025].

Su utilización directa en el ámbito sanitario presenta, no obstante, limitaciones relevantes. Los modelos pueden generar afirmaciones incorrectas con apariencia de veracidad, dependen del conocimiento incluido durante su entrenamiento y no garantizan acceso a documentación actualizada o específica de una organización. Además, las respuestas generadas únicamente a partir del conocimiento interno del modelo dificultan la trazabilidad de la información, lo que motiva el uso de mecanismos que permitan incorporar fuentes externas verificables.

3.2. Generación aumentada por recuperación

La generación aumentada por recuperación, conocida como RAG, combina un modelo generativo con un mecanismo de recuperación de información externa [Lewis et al., 2020]. Su objetivo es proporcionar al modelo contexto documental específico para cada consulta y conservar la procedencia de la información utilizada.

De forma general, el proceso consta de dos fases. Durante la indexación, los documentos se dividen en fragmentos y se representan mediante *embeddings*, que se almacenan junto con sus metadatos. Durante la consulta, la pregunta del usuario se representa mediante el mismo modelo y se recuperan los fragmentos más relevantes, que se incorporan al contexto proporcionado al LLM.

Este enfoque resulta especialmente útil en el ámbito sanitario porque permite trabajar con documentación que no forma parte del conocimiento original del modelo y facilita la trazabilidad de las respuestas. Sin embargo, RAG no garantiza por sí mismo la corrección del resultado: si la recuperación selecciona fragmentos irrelevantes o incompletos, la información proporcionada al modelo será insuficiente.

La revisión de Amugongo et al. sobre RAG en aplicaciones sanitarias muestra una considerable heterogeneidad tanto en las arquitecturas utilizadas como en los procedimientos de evaluación y destaca la ausencia de marcos estandarizados para este tipo de sistemas [Amugongo et al., 2025]. En este trabajo, RAG se utiliza por tanto como mecanismo de contextualización y trazabilidad, no como garantía absoluta de corrección.

3.3. Realidad Mixta aplicada a salud y telemedicina

La MR permite superponer información digital sobre el entorno físico manteniendo la percepción del mundo real, y se sitúa dentro del continuo realidad-virtualidad descrito por Milgram y Kishino [Milgram and Kishino, 1994]. En el ámbito sanitario, esta capacidad permite presentar instrucciones, paneles o información contextual sin depender exclusivamente de una pantalla convencional.

Worlikar et al. identificaron aplicaciones de realidad aumentada, virtual y mixta en ámbitos como teleconsulta, telementoría, telemonitorización, telerehabilitación y telediagnóstico, destacando el valor del punto de vista en primera persona y de las anotaciones espaciales [Worlikar et al., 2023]. Asimismo, Martin et al. utilizaron HoloLens 2 para facilitar la colaboración remota entre profesionales durante la pandemia de COVID-19 [Martin et al., 2020].

En el presente trabajo, la MR se utiliza para mantener accesibles las respuestas del asistente y la navegación de protocolos dentro del entorno físico del usuario. Esta integración introduce requisitos adicionales relacionados con legibilidad, disposición espacial e interacción con los elementos virtuales.

3.4. Asistentes clínicos basados en Inteligencia Artificial

Una de las líneas con mayor implantación reciente es la denominada IA ambiental, orientada principalmente a capturar conversaciones clínicas y generar documentación de forma automática. Duggan et al. estudiaron este tipo de herramientas en un entorno ambulatorio y observaron mejoras relacionadas con el tiempo dedicado a documentación y la carga percibida por los profesionales [Duggan et al., 2025].

En el ámbito comercial, Microsoft Dragon Copilot constituye un ejemplo de asistente integrado en el flujo de trabajo clínico, combinando captura de conversaciones, dictado, generación de notas y consulta mediante lenguaje natural [Microsoft, 2026].

Estas soluciones reflejan una tendencia hacia asistentes integrados en la actividad clínica, aunque su finalidad se centra principalmente en documentación y productividad. El presente trabajo se diferencia por combinar consulta documental, persistencia contextual y MR dentro del entorno MRP-5G.

3.5. Marco regulatorio

La utilización de IA en el ámbito sanitario debe considerar diferentes marcos normativos relacionados tanto con los propios sistemas de IA como con el tratamiento de información clínica.

El Reglamento (UE) 2024/1689, conocido como Reglamento de Inteligencia Artificial de la UE (AI Act), establece un enfoque basado en niveles de riesgo. Su artículo 6 determina que determinados sistemas de IA pueden clasificarse como sistemas de alto riesgo cuando forman parte de productos regulados por legislación armonizada de la Unión y están sujetos a una evaluación de conformidad por terceros [European Union, 2024].

Por tanto, la clasificación de un sistema sanitario basado en IA no depende únicamente de que utilice IA o de que exista supervisión humana. Deben considerarse factores como su finalidad prevista, su función concreta y su posible consideración como producto o componente de un producto sanitario.

El sistema desarrollado en este TFM constituye un prototipo experimental de apoyo y no se presenta como producto sanitario ni como sistema autónomo de diagnóstico o prescripción. La determinación formal de una posible clasificación regulatoria en un escenario de despliegue real queda fuera del alcance del trabajo.

También debe considerarse el Reglamento General de Protección de Datos (RGPD), Reglamento (UE) 2016/679. Los datos relativos a la salud se incluyen dentro de las categorías especiales de datos personales y, por tanto, están sujetos a requisitos específicos de protección y tratamiento [European Union, 2016].

Este aspecto resulta especialmente relevante para sistemas que puedan manejar información clínica real y refuerza la importancia de adoptar mecanismos adecuados de control de acceso, minimización de datos y protección de la información. En el prototipo desarrollado, el modelo de lenguaje se ejecuta sobre infraestructura controlada por la UCLM, evitando que las consultas deban enviarse a una API comercial externa para realizar la inferencia.

3.6. El proyecto MRP-5G

El presente TFM se desarrolla dentro del proyecto MRP-5G, una propuesta de investigación de la UCLM orientada al desarrollo de una plataforma de MR para soporte y diagnóstico remoto en atención primaria [García et al., 2025a, García et al., 2025b].

La propuesta integra tecnologías de MR, comunicaciones 5G e IA con el objetivo de facilitar la colaboración en tiempo real entre profesionales de atención primaria y médicos especialistas. Entre las funcionalidades planteadas se encuentran la videoconferencia en tiempo real, la incorporación de anotaciones virtuales, la gestión segura de información y la indexación inteligente de sesiones con fines de formación y consulta posterior.

La arquitectura propuesta es modular y escalable y diferencia varias capas funcionales destinadas a comunicaciones, persistencia, infraestructura, IA y comunicación en tiempo real. Dentro de la capa de IA se contempla específicamente la incorporación de asistentes conversacionales basados en LLMs capaces de ayudar al profesional a recuperar información técnica durante la atención al paciente.

El proyecto plantea tres escenarios principales de utilización: centros de atención primaria, atención en el domicilio del paciente y situaciones de emergencia. En todos ellos, el objetivo es permitir que el profesional que se encuentra físicamente junto al paciente pueda recibir apoyo remoto de un especialista mediante un dispositivo de MR y una infraestructura de comunicaciones de baja latencia.

Entre los objetivos definidos para la plataforma se encuentran el desarrollo de una aplicación de MR capaz de integrar información virtual y monitorización del paciente, una aplicación para especialistas remotos, la grabación e indexación inteligente de sesiones y una validación inicial del sistema en entornos controlados y centros sanitarios [García et al., 2025a].

Dentro de esta visión global, el presente TFM se centra específicamente en la vertiente de asistencia inteligente: el acceso mediante lenguaje natural a documentación clínica, la gestión contextual de las interacciones y la presentación de información y protocolos mediante una aplicación de MR.

3.7. Trabajo previo en MRP-5G

El antecedente más próximo al presente trabajo es el Trabajo Fin de Grado (TFG) de Pablo del Hoyo Abad, titulado *Intelligent system based on mixed reality for diagnosis and assistance in primary care*, desarrollado en la UCLM en 2024 [del Hoyo Abad, 2024].

El sistema desarrollado en dicho trabajo combinaba una aplicación web, un servicio para la gestión de protocolos, un servicio basado en un LLM y una aplicación de MR para Meta Quest 3.

La aplicación web permitía definir protocolos médicos en forma de diagramas de flujo y asociar documentación a dichos protocolos. Esta información podía utilizarse posteriormente desde la aplicación de MR para guiar al profesional a través de los distintos pasos definidos.

El servicio de IA incorporaba ya una arquitectura RAG. Los documentos se dividían en fragmentos y se representaban mediante *embeddings* generados con el modelo `BAAI/bge-large-en-v1.5`, denominado en adelante *bge-large-en*. Estos vectores se almacenaban en Qdrant y eran posteriormente recuperados en función de su similitud con la consulta. Para la generación se empleaba Mistral 7B y la librería LlamaIndex proporcionaba gran parte de la funcionalidad relacionada con la indexación, recuperación y generación.

La aplicación de Meta Quest 3 permitía ejecutar los protocolos definidos previamente y consultar al modelo de lenguaje durante su ejecución. Además, incorporaba reconocimiento y síntesis de voz mediante servicios de Wit.ai y permitía utilizar seguimiento de manos para determinadas interacciones.

Sin embargo, la interfaz web utilizada para consultar el modelo no mantenía una conversación persistente. El propio autor explica que las preguntas anteriores no se almacenaban ni se utilizaban posteriormente como contexto, por lo que el comportamiento se aproximaba más a consultas independientes que a una sesión conversacional completa.

Como puede observarse en la Figura 3.1, el proyecto MRP-5G integra tanto componentes web de soporte clínico remoto como escenarios de interacción en MR.

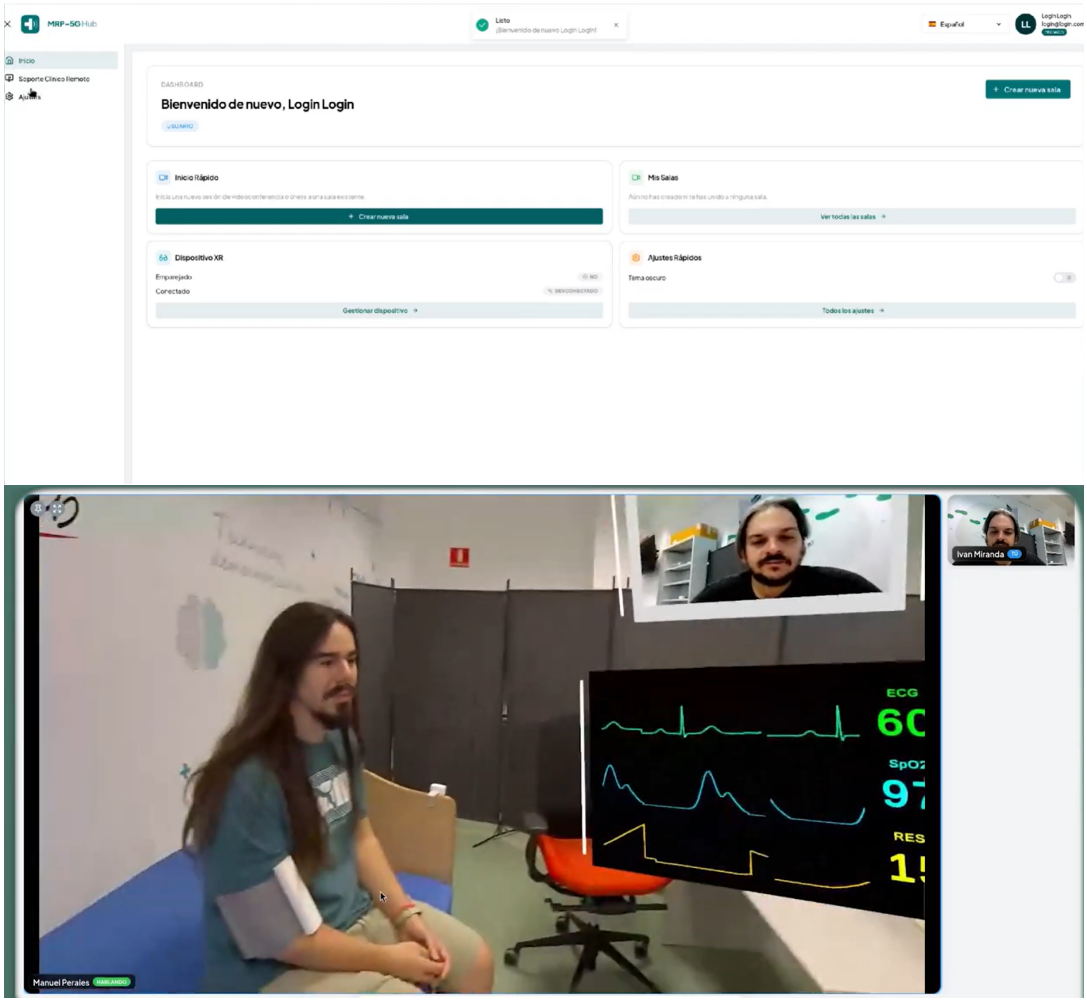


Figura 3.1: Ejemplos de interfaces y escenarios de uso de la plataforma MRP-5G. En la parte superior se muestra la interfaz web de gestión y acceso a salas de soporte clínico remoto. En la parte inferior se presenta un escenario de interacción en MR dentro del ecosistema del proyecto.

En las líneas de trabajo futuro del TFG se plantean, entre otras posibilidades, la validación del sistema con profesionales de atención primaria, el soporte para múltiples usuarios y consultas concurrentes, la exploración de alternativas a LlamaIndex y la incorporación de nuevas formas de interacción disponibles en Meta Quest 3.

Este trabajo constituye, por tanto, el antecedente conceptual y tecnológico más próximo al presente TFM. No obstante, la implementación realizada en este trabajo ha sido desarrollada de forma independiente.

3.8. Posicionamiento del presente TFM

El análisis anterior muestra que las tecnologías fundamentales utilizadas en este TFM no constituyen por separado una novedad. Los LLMs se están incorporando progresivamente a herramientas sanitarias, los sistemas RAG permiten contextualizar las respuestas utilizando información documental y la MR se ha investigado previamente como mecanismo de apoyo en telemedicina.

Asimismo, el trabajo previo desarrollado en el marco de MRP-5G ya demostró la viabilidad de combinar protocolos clínicos, recuperación documental, modelos de lenguaje y una aplicación de MR dentro de un mismo entorno tecnológico.

En este contexto, el presente trabajo se sitúa en la integración de dichas tecnologías dentro de un asistente con mayor continuidad entre interacciones y con mecanismos que van más allá de una consulta aislada.

Por tanto, la aportación del presente TFM no consiste en introducir de forma aislada RAG, interacción por voz o MR, tecnologías ya presentes en trabajos anteriores, sino en diseñar e implementar de manera independiente un asistente contextual y proactivo que integra estas capacidades dentro de una arquitectura común orientada a atención primaria.

4. Metodología

En este capítulo se describe la metodología de trabajo seguida durante el desarrollo del presente TFM. Se presenta el enfoque iterativo e incremental adoptado, se describen las principales tecnologías utilizadas y los medios hardware empleados y, finalmente, se explica cómo se ha aplicado dicha metodología a lo largo de las diferentes fases del proyecto.

4.1. Enfoque metodológico

El desarrollo del proyecto se ha organizado mediante una adaptación ligera de Scrum [Schwaber and Sutherland, 2020], para la gestión del trabajo y un enfoque iterativo e incremental basado en prototipado evolutivo [Larman and Basili, 2003] para el desarrollo software. Al tratarse de un TFM individual, no se han aplicado de forma estricta los roles, eventos ni ciclos temporales de Scrum, sino sucesivas iteraciones funcionales revisadas periódicamente con los directores del trabajo.

Este enfoque ha permitido ajustar el alcance y las prioridades en función de los resultados obtenidos, integrar progresivamente los distintos componentes y mantener versiones funcionales durante el desarrollo. Las principales decisiones técnicas se describen en el Capítulo 5.

Estas iteraciones no correspondieron a *sprints* de duración fija, sino a bloques de trabajo definidos en función de los principales objetivos técnicos del proyecto. El trabajo se estructuró en cinco iteraciones principales:

- **Primera iteración – prototipo inicial y sistema RAG**: se implementó una primera versión funcional formada por el *backend*, el procesamiento e indexación documental, la recuperación semántica y una interfaz web básica capaz de realizar consultas al modelo.
- **Segunda iteración – sesiones y contextualización**: se incorporó persistencia mediante PostgreSQL, gestión de sesiones e historial conversacional, asociación de documentos al contexto activo y mecanismos proactivos como preguntas sugeridas y extracción de tareas. También se integró la entrada mediante voz.
- **Tercera iteración – autenticación, seguimiento y simulación**: se añadieron los enlaces de acceso mediante correo electrónico, las preferencias de usuario, los recordatorios de tareas y el simulador de casos clínicos ficticios.
- **Cuarta iteración – aplicación de MR**: se desarrolló la aplicación para Meta Quest 3, integrando la comunicación con el *backend*, la consulta mediante voz, los paneles espaciales y la representación interactiva del protocolo clínico.
- **Quinta iteración – integración y refinamiento**: se dedicó a integrar los componentes desarrollados, corregir problemas de interfaz e interacción, refinar el árbol de decisión clínico y realizar las pruebas funcionales y de rendimiento del prototipo.

Este proceso permitió disponer de versiones funcionales del sistema a lo largo del desarrollo y adaptar progresivamente el alcance en función de los resultados obtenidos y de las reuniones de seguimiento.

4.2. Marco tecnológico

La solución integra distintas tecnologías en función de las necesidades de cada componente. La aplicación web se ha desarrollado con Angular y TypeScript, mientras que el *backend* utiliza FastAPI y Python. PostgreSQL proporciona la persistencia de datos estructurados y Qdrant el almacenamiento y recuperación vectorial utilizados por el sistema RAG. Asimismo, Whisper se emplea para la transcripción de voz.

La aplicación de MR se ha desarrollado en Unity para Meta Quest 3 utilizando Meta XR. Git se emplea para el control de versiones y Docker para facilitar la contenerización y el despliegue de los principales componentes.

La Figura 4.1 resume de forma esquemática el conjunto de tecnologías empleadas en el desarrollo del sistema.



Figura 4.1: Resumen de las principales tecnologías y herramientas empleadas en el desarrollo del sistema.

4.3. Medios hardware

El desarrollo se ha realizado principalmente en un equipo personal con un **procesador**: Intel Core i7, una **memoria RAM**: 24 GB DDR5, una **tarjeta gráfica**: NVIDIA GeForce RTX 4050 y un **sistema operativo**: Windows 11. Para el desarrollo y las pruebas de la aplicación de MR se ha utilizado un dispositivo Meta Quest 3.

El modelo de lenguaje `gpt-oss:20b` se ejecuta en un servidor del grupo de investigación AIR de la UCLM, al que el *backend* accede a través de la red de la Universidad. De esta forma, los componentes de la aplicación se ejecutan localmente, mientras que la inferencia se delega en una infraestructura con capacidad suficiente para alojar el modelo.

4.4. Planificación temporal

La Figura 4.2 muestra la distribución de las principales fases del proyecto. La planificación se utilizó como referencia y se fue ajustando durante las reuniones periódicas con los directores en función de los avances y necesidades aparecidas durante el desarrollo.

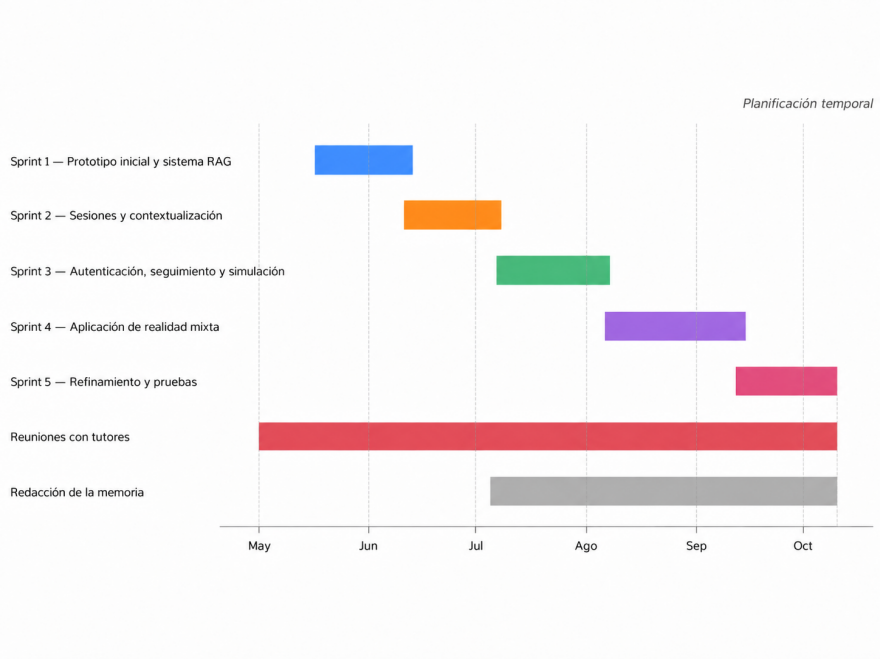


Figura 4.2: Planificación temporal del proyecto

4.5. Organización del código fuente

El código fuente se encuentra centralizado en un único repositorio público bajo control de versiones mediante Git. El repositorio `https://github.com/chemabr24/tfm-llm-assistant-mrp5g` contiene el *backend* desarrollado con FastAPI, la aplicación web en Angular y la aplicación de MR desarrollada con Unity para Meta Quest 3:

5. Arquitectura

En este capítulo se presenta la arquitectura diseñada para el sistema y las principales decisiones adoptadas durante su desarrollo. El objetivo no es describir únicamente el funcionamiento de cada componente, sino justificar cómo se ha estructurado la solución, qué problemas han condicionado su diseño, qué alternativas se han considerado y qué ventajas proporciona la solución adoptada.

La arquitectura se analiza a partir de sus principales módulos: las interfaces de usuario, el núcleo de servicios, los mecanismos de persistencia y recuperación de información, el sistema RAG y la aplicación de MR. Para cada uno de ellos se describen las decisiones de diseño más relevantes y su relación con las tecnologías utilizadas.

5.1. Visión general de la arquitectura del sistema

La arquitectura del sistema se ha diseñado siguiendo un enfoque modular, separando la interacción con el usuario, la lógica de aplicación, la persistencia de información, la recuperación documental y la generación mediante el modelo de lenguaje. Esta separación responde principalmente a la necesidad de proporcionar las mismas capacidades del asistente desde clientes con características muy diferentes: una aplicación web y una aplicación de MR.

La Figura 5.1 muestra la organización conceptual de la solución. En lugar de implementar una lógica independiente para cada cliente, ambas interfaces acceden a un núcleo común de servicios que centraliza la gestión de usuarios y sesiones, el procesamiento documental,

la recuperación semántica, la comunicación con el LLM, la transcripción de voz y las funcionalidades de seguimiento.

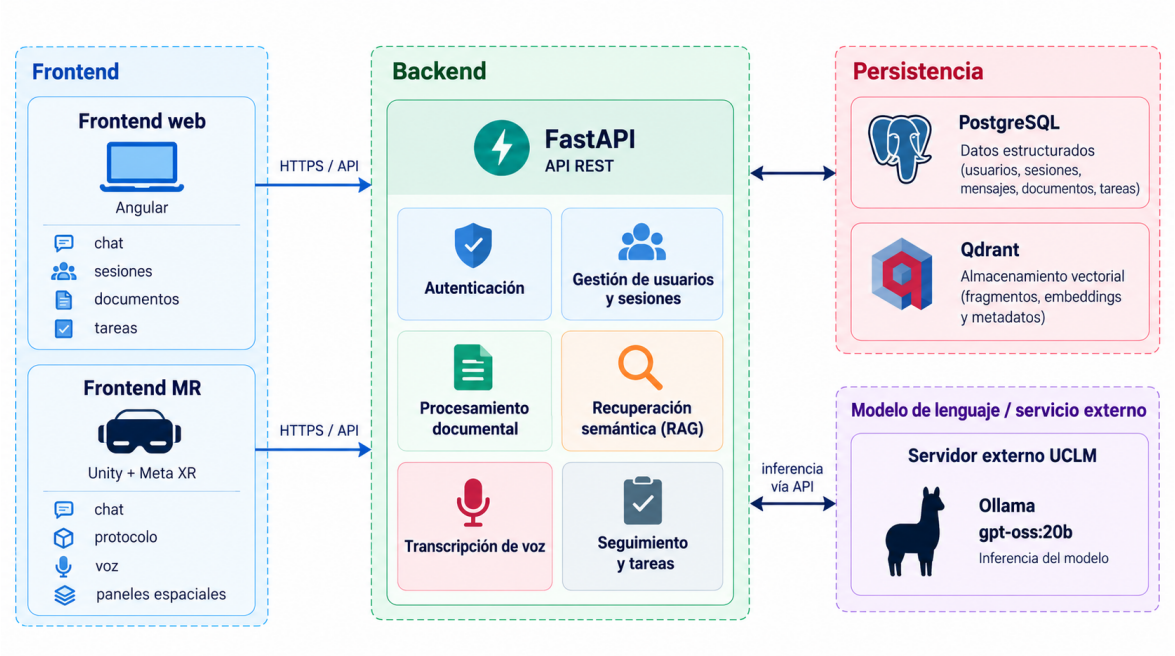


Figura 5.1: Visión general de la arquitectura del sistema y relación entre sus principales componentes.

La utilización de un núcleo común evita duplicar lógica entre las interfaces web y de MR y permite que ambas compartan la misma información y el mismo contexto de trabajo. De esta forma, documentos, conversaciones y otros elementos asociados a una sesión permanecen centralizados independientemente del cliente desde el que se acceda al sistema.

La persistencia se divide conceptualmente en dos mecanismos debido a la distinta naturaleza de la información gestionada. Los datos estructurados, como usuarios, sesiones, mensajes, documentos y tareas, requieren relaciones y operaciones propias de una base de datos relacional. En cambio, los fragmentos documentales deben poder recuperarse atendiendo a su similitud semántica. Esta diferencia motivó la separación entre persistencia relacional y almacenamiento vectorial.

El componente de recuperación documental actúa como intermediario entre la información incorporada al sistema y el LLM. Las consultas no se envían directamente al modelo, sino que previamente se recupera información relevante correspondiente exclusiva-

mente al contexto de trabajo activo. Esta decisión permite aplicar una arquitectura RAG [Lewis et al., 2020], proporcionando al modelo un contexto documental explícito y manteniendo la trazabilidad de la información utilizada.

Sobre esta arquitectura conceptual se seleccionaron las tecnologías concretas. Angular se utiliza para la interfaz web, Unity y Meta XR para la aplicación de MR, FastAPI para implementar el núcleo de servicios, PostgreSQL para la persistencia relacional y Qdrant para el almacenamiento y recuperación de representaciones vectoriales. El modelo `gpt-oss:20b`, desplegado en infraestructura de la UCLM, constituye el componente encargado de la generación de lenguaje.

5.2. Backend y persistencia

La Figura 5.2 muestra la organización interna del *backend*, diferenciando los puntos de entrada de la API, los servicios que encapsulan la lógica de aplicación y los mecanismos de persistencia y servicios externos utilizados.

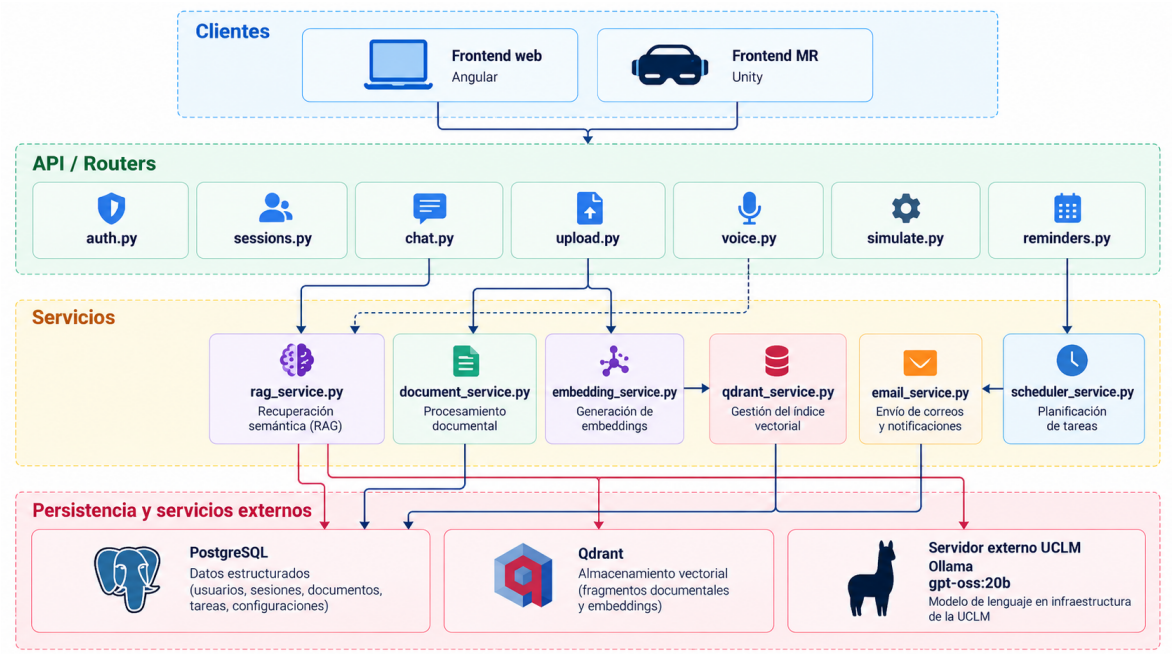


Figura 5.2: Organización por capas del *backend* y relación entre *routers*, servicios, persistencia y servicios externos.

5.2.1. Organización del backend

El sistema debía proporcionar un conjunto común de funcionalidades a dos clientes con características diferentes: la aplicación web y la aplicación de MR. Implementar de forma independiente en cada cliente la gestión de sesiones, documentos, recuperación de información o comunicación con el modelo habría provocado duplicación de lógica y un mayor acoplamiento entre las interfaces y los servicios utilizados.

Para abordar este problema se centralizó la lógica de aplicación en un *backend* desarrollado con FastAPI. Como se muestra en la Figura 5.2, la implementación separa los *routers*, encargados de exponer las operaciones de la API, de los servicios que encapsulan funcionalidades específicas. Los *routers* gestionan operaciones relacionadas con autenticación, sesiones, conversación, documentos, voz, simulación y recordatorios, mientras que servicios como `RAGService`, `DocumentService`, `EmbeddingService` o `QdrantService` concentran la lógica reutilizable asociada a cada funcionalidad.

Esta organización permite que ambos clientes consuman las mismas operaciones sin conocer los detalles internos de persistencia, recuperación o generación. Además, la separación entre la capa de acceso y los servicios facilita modificar componentes concretos, realizar pruebas de forma aislada y mantener una única implementación de la lógica compartida.

5.2.2. Modelo basado en sesiones

Una de las principales decisiones arquitectónicas ha sido organizar el sistema alrededor del concepto de sesión. Cada sesión representa un contexto de trabajo independiente al que se asocian los documentos, mensajes, tareas y, de forma opcional, un identificador de paciente.

Esta organización responde a la necesidad de evitar que información perteneciente a contextos diferentes se mezcle durante la recuperación o la generación. El identificador de sesión se utiliza tanto en la persistencia relacional como durante las búsquedas vectoriales, de forma que una consulta solo pueda recuperar información perteneciente al contexto activo.

Se han definido cinco entidades principales:

- **Usuario**: almacena la información básica de la cuenta y sus preferencias.

- **Sesión**: representa un contexto independiente de conversación y documentación.
- **Documento**: identifica cada recurso incorporado a una sesión.
- **Mensaje**: almacena las intervenciones del usuario y del asistente, incluyendo las fuentes asociadas a las respuestas.
- **Tarea**: representa acciones identificadas durante las interacciones y vinculadas a una sesión concreta.

Las relaciones entre las principales entidades persistentes se representan en la Figura 5.3. La sesión actúa como elemento central al que se vinculan los documentos, mensajes y tareas generados durante la interacción.

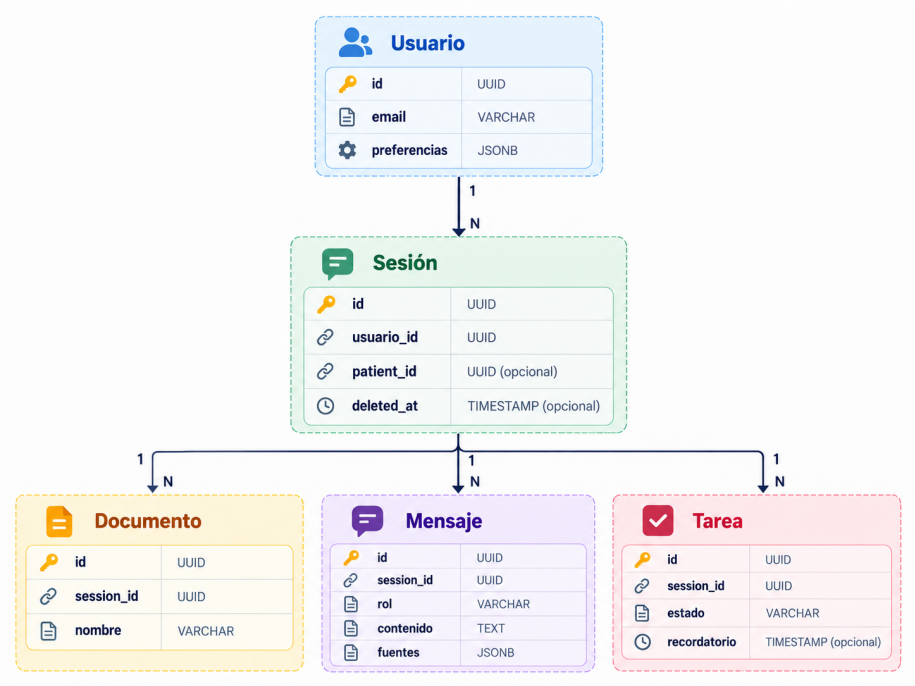


Figura 5.3: Relaciones principales entre las entidades utilizadas en el modelo basado en sesiones.

La eliminación de sesiones se realiza mediante *soft delete*. Esta estrategia permite excluir una sesión del uso habitual sin eliminar inmediatamente los registros asociados, conservando la posibilidad de mantener la integridad de la información almacenada durante el desarrollo y las pruebas.

5.2.3. Persistencia relacional y vectorial

Como se muestra en la Figura 5.2, la capa de persistencia se divide entre PostgreSQL y Qdrant, ya que ambos componentes resuelven necesidades diferentes.

PostgreSQL se utiliza para almacenar información estructurada y relacionada, como usuarios, sesiones, mensajes, documentos y tareas. Este tipo de información presenta relaciones bien definidas y requiere operaciones de consulta y actualización convencionales.

Qdrant, por el contrario, se utiliza para almacenar los vectores asociados a los fragmentos documentales. La elección de un almacén vectorial independiente permite realizar búsquedas por similitud semántica sin trasladar esta responsabilidad a la base de datos relacional.

Aunque esta separación incrementa el número de componentes del sistema, aporta una delimitación clara de responsabilidades y permite que cada tecnología se utilice para el tipo de información para el que resulta más adecuada.

5.2.4. Servicios compartidos

Además de los servicios relacionados directamente con recuperación y persistencia, la arquitectura incorpora varias funcionalidades compartidas entre los clientes. Como se observa en la Figura 5.2, estas capacidades se exponen mediante *routers* específicos y reutilizan servicios internos del *backend*.

Para el acceso a la aplicación se eligió un mecanismo basado en *magic links* enviados mediante correo electrónico.

Esta solución evita introducir en el prototipo un sistema completo de gestión de contraseñas y recuperación de credenciales. El servidor genera un *token* de un solo uso con una validez temporal y verifica su estado antes de permitir el acceso.

La solución resulta adecuada para el alcance del prototipo, aunque en un despliegue real debería integrarse con mecanismos de autenticación institucionales o con una infraestructura específica de gestión de identidad.

La transcripción se ha centralizado en el *backend* mediante Whisper en lugar de implementar mecanismos diferentes para cada cliente.

Esta decisión permite que tanto la aplicación web como la aplicación de MR reutilicen el mismo servicio y evita depender de soluciones específicas del navegador o del fabricante del dispositivo.

En la aplicación web, el texto obtenido mediante la transcripción se introduce primero en el campo de consulta para que pueda ser revisado antes de enviarse al asistente. Esta separación evita utilizar automáticamente una transcripción potencialmente incorrecta como entrada para el modelo.

El simulador de casos se diseñó como un mecanismo para generar escenarios ficticios y proporcionar material de prueba sin depender de datos clínicos reales.

El caso generado se organiza en diferentes apartados clínicos y se transforma posteriormente en un documento de texto asociado a la sesión. Una vez creado, este documento utiliza el mismo pipeline de indexación que cualquier otro recurso incorporado al sistema.

La reutilización del mecanismo general de indexación evita mantener un flujo específico para los casos simulados y permite tratarlos posteriormente como una fuente documental más durante las consultas.

Durante el desarrollo se descartó almacenar estos informes incluyendo etiquetas HTML y sintaxis Markdown, ya que dificultaban su lectura como documentos independientes. La solución final utiliza texto plano estructurado.

Los nombres de los documentos incorporan un identificador legible y un UUID, evitando colisiones cuando diferentes sesiones utilizan títulos o identificadores similares.

Las tareas generadas a partir de las conversaciones se almacenan como entidades persistentes y pueden clasificarse como pendientes, completadas o archivadas.

Sobre esta información se incorporó un mecanismo de recordatorios mediante correo electrónico. La planificación se realiza mediante APScheduler y distingue entre frecuencias diaria y semanal.

El envío solo se produce cuando existen tareas pendientes. Esta decisión evita generar notificaciones sin información relevante y mantiene el mecanismo de recordatorio vinculado al estado real de las sesiones del usuario.

La implementación actual ejecuta el planificador dentro del mismo proceso del *backend*. Esta solución simplifica el prototipo, aunque introduce una dependencia de la disponibilidad continua del servicio que debería revisarse en un despliegue de producción.

5.3. Arquitectura de recuperación y generación

5.3.1. Procesamiento documental

La documentación incorporada al sistema debe transformarse antes de poder ser utilizada como contexto por el asistente. Para los documentos PDF se conserva la procedencia por página durante la extracción del texto, ya que posteriormente esta información permite mostrar al usuario el origen de los fragmentos recuperados.

El contenido se divide en fragmentos de aproximadamente 512 palabras con un solapamiento de 50 palabras entre fragmentos consecutivos. El objetivo del solapamiento es disminuir la posibilidad de perder información contextual cuando una idea queda situada en el límite entre dos fragmentos.

Cada fragmento conserva, además del texto, diferentes metadatos:

- identificador único;
- página de procedencia;
- nombre del documento;
- identificador de la sesión.

La conservación de estos metadatos responde a dos necesidades. Por una parte, el identificador de sesión permite aislar los contextos documentales. Por otra, el nombre del documento y la página permiten mantener trazabilidad sobre la información recuperada.

5.3.2. Representación y recuperación semántica

Los fragmentos documentales se representan mediante *embeddings* generados con el modelo `intfloat/multilingual-e5-large` [Wang et al., 2024].

La elección final de un modelo multilingüe responde a la necesidad de trabajar principalmente con consultas y documentación en español. Las representaciones generadas se normalizan y se almacenan en Qdrant, utilizando similitud coseno durante el proceso de recuperación.

El almacén vectorial conserva conjuntamente el vector y los metadatos del fragmento. Esto permite combinar la similitud semántica con filtros estructurados, especialmente el

identificador de la sesión activa. Estas operaciones corresponden a las etapas de representación y recuperación mostradas en la Figura 5.4.

5.3.3. Diseño del pipeline RAG

El mecanismo de consulta se ha diseñado siguiendo una arquitectura RAG, propuesta por Lewis et al. [Lewis et al., 2020]. Su utilización responde a la necesidad de reducir la dependencia del conocimiento interno del modelo y proporcionar información documental explícita sobre la que fundamentar las respuestas.

La Figura 5.4 representa conceptualmente las principales etapas del proceso de recuperación y generación utilizado por el sistema.

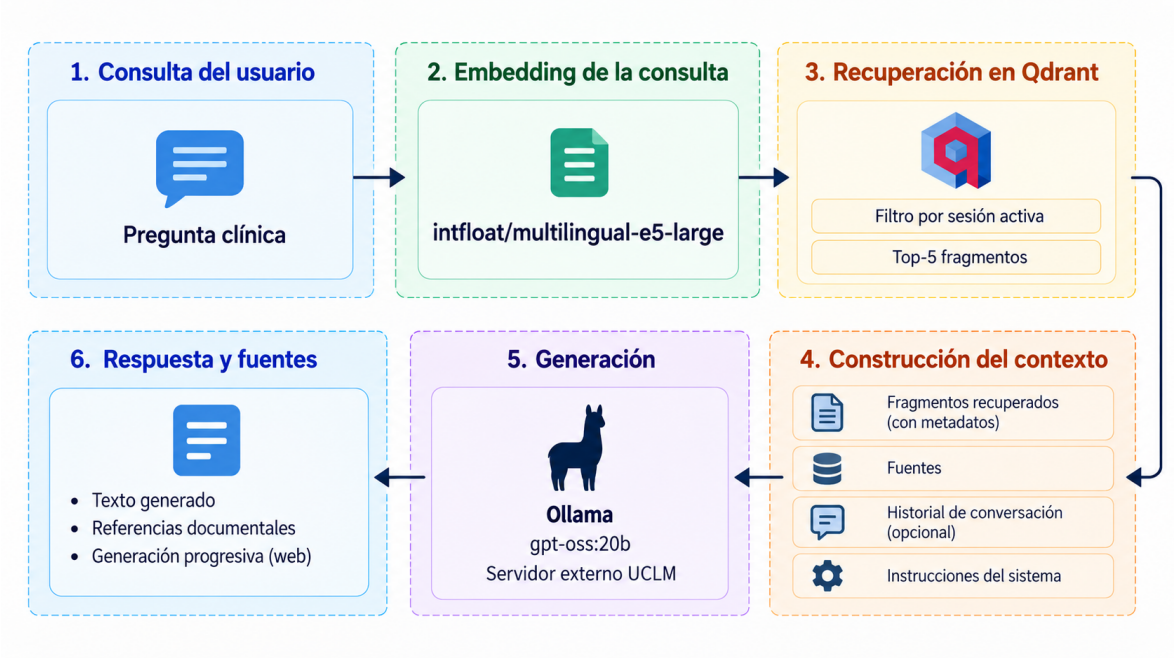


Figura 5.4: Etapas del proceso de recuperación de información y generación de respuestas mediante RAG.

Ante una consulta, el sistema genera su representación vectorial y recupera hasta cinco fragmentos considerados semánticamente próximos. La búsqueda se restringe mediante el identificador de la sesión activa, evitando utilizar documentación correspondiente a otros contextos.

Los fragmentos recuperados se incorporan posteriormente al contexto enviado al LLM junto con su documento y página de procedencia. El *prompt* del sistema establece además restricciones orientadas a que las respuestas se apoyen en la documentación proporcionada, indiquen cuándo la información no está disponible y eviten realizar diagnósticos o prescripciones de forma autónoma.

La elección de separar explícitamente recuperación y generación permite inspeccionar qué documentación ha sido seleccionada antes de generar la respuesta y conservar las fuentes asociadas al resultado.

5.3.4. Contexto conversacional y trazabilidad

El historial asociado a la sesión puede incorporarse a nuevas consultas con el objetivo de mantener continuidad entre intervenciones relacionadas, formando parte del contexto representado en la etapa 4 de la Figura 5.4. El sistema no trata cada consulta como una interacción completamente independiente.

La persistencia del historial en PostgreSQL permite reconstruir posteriormente la conversación desde la aplicación web, mientras que la asociación de las fuentes a cada mensaje permite conservar la procedencia documental de las respuestas.

Es importante diferenciar esta trazabilidad de una medida de confianza clínica. La puntuación obtenida durante la recuperación representa únicamente similitud semántica entre la consulta y los fragmentos almacenados, y no una estimación de la corrección médica de la respuesta.

5.3.5. Generación progresiva y funcionalidades proactivas

Las etapas finales mostradas en la Figura 5.4 comprenden la generación de la respuesta y la devolución de las fuentes asociadas. En la aplicación web se optó además por transmitir progresivamente la respuesta generada en lugar de esperar a que el modelo completase todo el contenido.

Esta decisión no reduce necesariamente el tiempo total de generación, pero permite disminuir la percepción de espera al mostrar texto al usuario desde que comienzan a recibirse los primeros fragmentos.

El *backend* consume en modo *streaming* la respuesta del servicio del modelo y la transmi-

te al cliente mediante una respuesta HTTP incremental. En el navegador, esta información se procesa mediante la API *Fetch* y un flujo de lectura progresivo.

Las fuentes se incorporan al finalizar la generación como información independiente del cuerpo principal de la respuesta. Esta separación simplifica su tratamiento en las interfaces y evita mezclar las referencias documentales con el texto generado por el modelo.

El asistente se diseñó con el objetivo de ir más allá de una interacción puramente reactiva basada en preguntas y respuestas. Para ello se incorporaron mecanismos como la generación de preguntas sugeridas, la identificación automática de tareas y la generación de información inicial al incorporar determinados documentos.

Estas funciones se realizan mediante operaciones independientes de la generación principal. Se descartó concentrar todas las salidas en una única respuesta del modelo, ya que separar cada operación permite utilizar instrucciones específicas, controlar mejor el formato esperado y evitar que información destinada a tareas o sugerencias se mezcle con el contenido principal de la respuesta.

En el caso de la extracción de tareas, se utiliza una llamada específica al modelo solicitando una estructura JSON. Las tareas identificadas se almacenan posteriormente en PostgreSQL y permanecen asociadas a la sesión correspondiente.

Este diseño implica realizar llamadas adicionales al modelo, pero proporciona un mayor control sobre cada funcionalidad y simplifica su tratamiento posterior por parte de la aplicación.

5.4. Frontend web

La aplicación web actúa como interfaz principal de gestión del sistema y se ha desarrollado utilizando Angular 22 y TypeScript.

A diferencia de una interfaz orientada exclusivamente al chat, su diseño se estructura alrededor del contexto representado por una sesión. Desde este entorno se integran la conversación, los documentos asociados y las tareas generadas durante las interacciones.

La interfaz se organiza conceptualmente en tres áreas:

- gestión de sesiones;
- conversación con el asistente;

- información asociada a la sesión, como documentos y tareas.

Esta separación pretende evitar concentrar toda la información en el área de conversación y mantener accesibles de forma independiente los elementos que persisten más allá de un mensaje concreto.

5.5. Frontend de realidad mixta

La segunda interfaz del sistema se ha diseñado específicamente para Meta Quest 3 utilizando Unity 2022.3 LTS y Meta XR.

A diferencia de la aplicación web, el diseño de esta interfaz debe considerar restricciones propias de la visualización espacial, como la distancia de lectura, el tamaño de los paneles, la interacción mediante controladores y la necesidad de no ocultar innecesariamente el entorno físico del usuario.

La Figura 5.5 muestra los principales componentes de la aplicación de MR, sus responsabilidades y las relaciones existentes entre ellos y el *backend*.

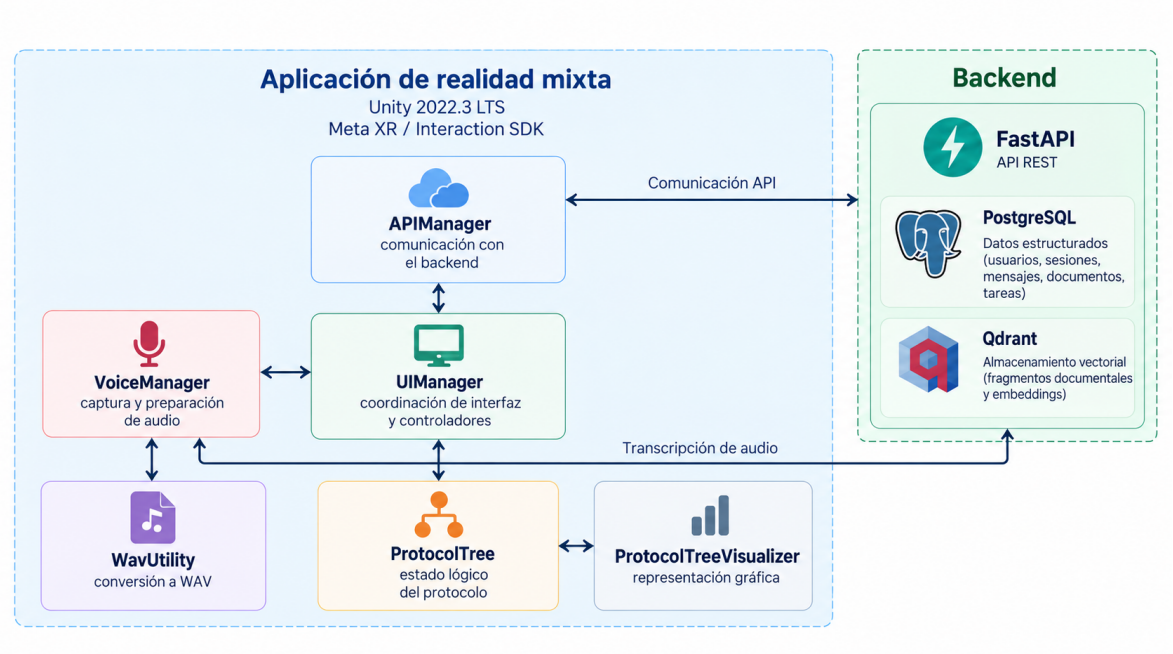


Figura 5.5: Componentes principales de la aplicación de MR y su comunicación con el *backend*.

5.5.1. Separación de responsabilidades

Como se representa en la Figura 5.5, la lógica de la aplicación se divide entre diferentes componentes con responsabilidades específicas:

- **APIManager**: comunicación con el *backend* y gestión de las consultas al asistente.
- **VoiceManager**: captura y preparación del audio.
- **UIManager**: coordinación de la interfaz y acciones asociadas a los controladores.
- **ProtocolTree**: representación del estado lógico del protocolo.
- **ProtocolTreeVisualizer**: generación de su representación gráfica.
- **WavUtility**: transformación del audio capturado al formato utilizado por el servicio de transcripción.

La separación entre `ProtocolTree` y `ProtocolTreeVisualizer` resulta especialmente relevante, ya que desacopla la estructura lógica del protocolo de su representación visual. De este modo, la presentación puede modificarse sin alterar las reglas que determinan la navegación entre nodos.

5.5.2. Comunicación con el backend

La comunicación representada entre `APIManager` y el *backend* en la Figura 5.5 permite que la aplicación de MR consuma la misma API que la aplicación web. Esta decisión mantiene una única implementación de la lógica de negocio y garantiza que ambos clientes operen sobre las mismas sesiones y mecanismos de persistencia.

En la implementación actual se utiliza `UnityWebRequest` junto con un `DownloadHandlerBuffer`. Como consecuencia, la aplicación espera a recibir completamente la respuesta antes de mostrarla.

Esta solución se eligió inicialmente por su menor complejidad durante la integración con Unity. A diferencia de la aplicación web, no proporciona visualización progresiva de la generación, por lo que la incorporación de un mecanismo de *streaming* equivalente constituye una posible mejora futura.

5.5.3. Interacción y paneles espaciales

Uno de los principales problemas encontrados durante el desarrollo fue conseguir una interacción estable entre los controladores de Meta Quest 3 y las interfaces creadas mediante *World Space Canvas*.

Durante las primeras pruebas se utilizaron componentes del sistema XR genérico de Unity y mecanismos de entrada procedentes de integraciones anteriores de Oculus. La combinación inicial no proporcionaba una interacción suficientemente fiable con los controles de la interfaz.

La solución final se construyó utilizando `OVRCameraRig` y Meta Interaction SDK. Sobre el *rig* se añadieron interactores de rayo para los controladores, mientras que los elementos de interfaz utilizan componentes compatibles con el sistema de puntero de Meta.

Esta decisión permitió unificar la interacción con botones, nodos del protocolo y paneles mediante un mismo mecanismo de selección.

Además del rayo, algunas acciones frecuentes se mantienen asociadas a botones físicos de los controladores, proporcionando una vía alternativa de interacción.

Las primeras versiones de la interfaz utilizaban paneles situados en posiciones predeterminadas. Durante las pruebas se observó que una disposición fija podía resultar incómoda dependiendo de la orientación y postura del usuario.

Por este motivo se incorporó la posibilidad de desplazar y reorganizar los paneles dentro del entorno de realidad mixta. Cada panel dispone de una zona de agarre que permite modificar su posición y orientación.

Los contenidos extensos utilizan componentes de desplazamiento, ya que tanto las conversaciones como la representación del protocolo pueden superar el área visible inicialmente.

La capacidad de reorganizar espacialmente la interfaz permite adaptar el entorno virtual a las necesidades de cada ejecución sin imponer una disposición única.

5.5.4. Protocolo clínico

El protocolo incorporado al prototipo se representa mediante una estructura de nodos conectados. Cada nodo contiene un identificador, un título, una descripción, un conjunto de opciones y la información necesaria para determinar si la rama puede continuar.

Cada opción referencia al siguiente nodo, permitiendo representar el protocolo como una estructura de decisión independiente de su visualización.

Esta separación entre datos y representación facilita modificar el aspecto de la interfaz sin tener que redefinir la lógica del procedimiento.

La primera aproximación utilizada durante el desarrollo mostraba principalmente la decisión correspondiente al estado actual. Aunque este diseño simplificaba la interfaz, proporcionaba poca información sobre el camino seguido y las alternativas disponibles en decisiones anteriores.

Por este motivo se adoptó finalmente una representación mediante árbol de decisión progresivo. Cuando el usuario selecciona una opción, el siguiente nivel se incorpora visualmente sin eliminar los niveles anteriores.

La solución mantiene visibles:

- el camino seleccionado hasta el estado actual;
- las alternativas disponibles en niveles anteriores;
- las opciones correspondientes al nodo activo.

Esta representación permite comprender visualmente cómo se ha alcanzado el estado actual y facilita modificar una decisión previa. Cuando el usuario selecciona una alternativa perteneciente a un nivel anterior, la rama posterior deja de ser válida y se reconstruye a partir de la nueva elección.

La separación entre el estado lógico almacenado en `ProtocolTree` y su representación mediante `ProtocolTreeVisualizer`, mostrada también en la Figura 5.5, permite realizar esta reconstrucción sin acoplar las reglas del protocolo a los elementos gráficos.

El protocolo y el asistente se han diseñado como mecanismos complementarios. El árbol proporciona una estructura determinista para recorrer un procedimiento, mientras que el asistente ofrece acceso a información documental relacionada con el estado actual.

Para conectar ambos mecanismos, el nodo seleccionado puede utilizarse como contexto para formular automáticamente una consulta al asistente. De esta forma, el usuario puede continuar siguiendo la estructura del protocolo y, al mismo tiempo, solicitar información adicional cuando resulta necesario.

5.6. Configuración y despliegue

La configuración dependiente del entorno se mantiene separada del código fuente mediante variables de entorno. Entre ellas se incluyen las conexiones a los servicios de persistencia, la configuración del correo electrónico y las credenciales de acceso al servicio que aloja el modelo de lenguaje. Los valores reales no se almacenan en el repositorio; en su lugar, se proporciona un fichero `.env.example` con las variables necesarias.

El sistema puede desplegarse mediante Docker Compose, que agrupa el *frontend*, el *backend*, PostgreSQL y Qdrant. Como muestra la Figura 5.6, el modelo de lenguaje se ejecuta de forma independiente en infraestructura de la UCLM y se consume como un servicio externo.

La separación en contenedores permite aislar las dependencias de los componentes y facilitar la reproducción del entorno. PostgreSQL y Qdrant utilizan volúmenes persistentes para conservar la información entre reinicios o reconstrucciones de los contenedores.

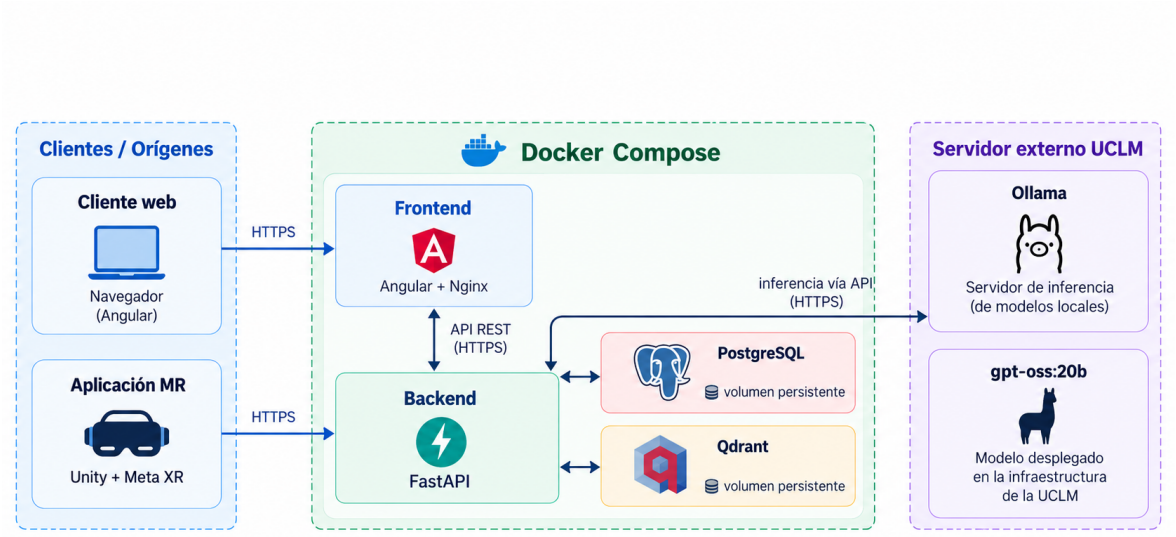


Figura 5.6: Distribución de los componentes del sistema y comunicación con el servicio de inferencia externo.

6. Resultados

En este capítulo se presentan los resultados obtenidos durante la evaluación del sistema desarrollado. El objetivo principal de las pruebas fue analizar el comportamiento del núcleo RAG, tanto de forma aislada en su etapa de recuperación documental como de forma completa, incluyendo la generación de respuestas mediante el modelo de lenguaje.

El código fuente y el vídeo demostrativo del prototipo se encuentran disponibles públicamente en los siguientes enlaces:

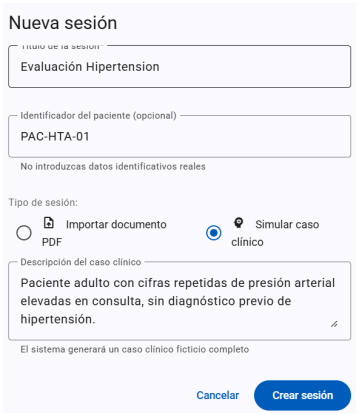
Repositorio: `https://github.com/chemabr24/tfm-llm-assistant-mrp5g`
Vídeo demostrativo: https://youtu.be/XA01EfJo3Hc

La evaluación se planteó desde una perspectiva técnica y funcional. Por tanto, los resultados presentados no constituyen una validación clínica del sistema, sino una evaluación de su capacidad para recuperar información relevante de documentación médica previamente incorporada, generar respuestas fundamentadas en dicha documentación y comportarse de forma adecuada cuando la evidencia disponible es insuficiente.

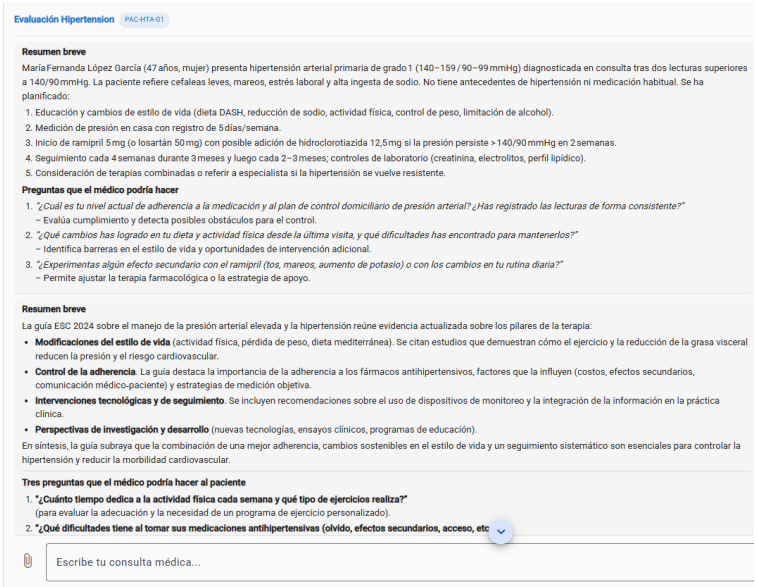
6.1. Resultados funcionales del prototipo

Antes de presentar la evaluación técnica del sistema, se muestran algunos de los principales resultados funcionales obtenidos en el prototipo desarrollado. Las siguientes capturas ilustran el flujo de creación de una sesión, la generación de información proactiva a partir del contexto disponible y la interacción con el asistente mediante recuperación documental.

La Figura 6.1a muestra la creación de una sesión orientada a la evaluación de un caso de hipertensión. En este ejemplo se utiliza un identificador ficticio y una descripción inicial del caso, a partir de la cual el sistema genera automáticamente un escenario clínico simulado.



(a) Creación de una sesión clínica simulada.



(b) Generación proactiva de información.

Figura 6.1: Creación de una sesión clínica simulada y generación proactiva de información asociada.

Una vez creada la sesión, el asistente puede generar información de forma proactiva. Como se observa en la Figura 6.1b, el sistema proporciona inicialmente un resumen del caso clínico simulado y posteriormente genera un nuevo resumen al incorporar documentación médica a la sesión. De esta forma, el usuario dispone de contexto inicial sin necesidad de formular una primera consulta manual.

La Figura 6.2 muestra un ejemplo completo de interacción con el asistente. La consulta se realiza dentro de una sesión que contiene tanto un caso clínico simulado como documentación médica de referencia. Además de la respuesta generada, la interfaz muestra los fragmentos documentales utilizados como fuentes, preguntas sugeridas para continuar la interacción y tareas extraídas automáticamente de la conversación.

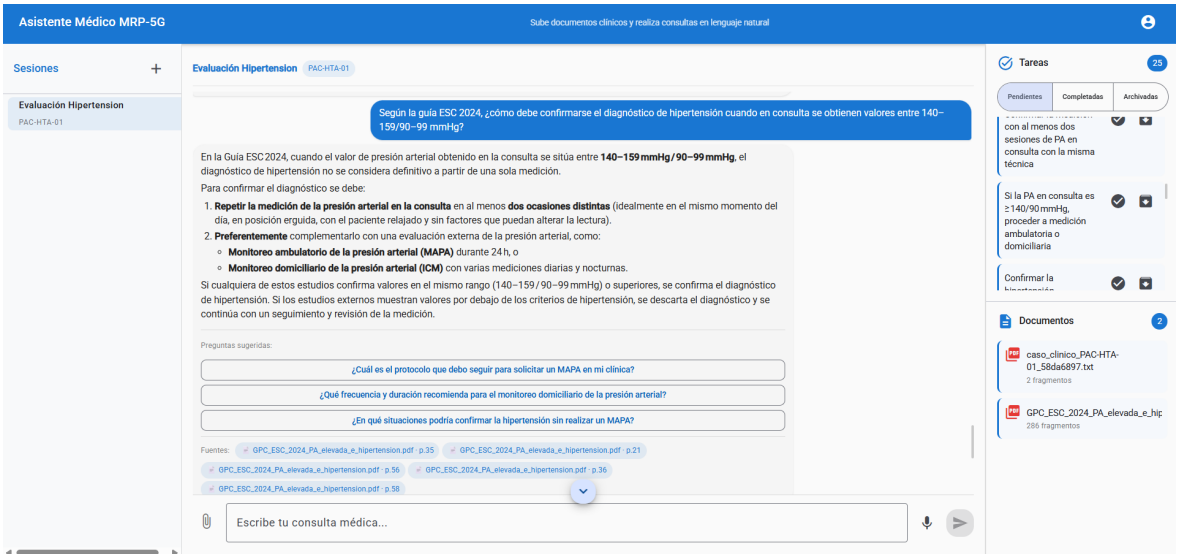
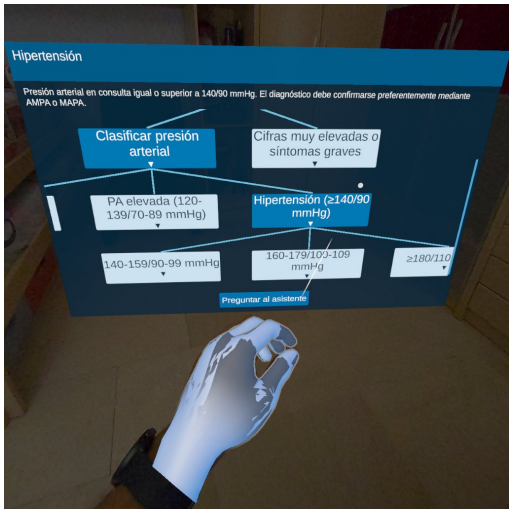


Figura 6.2: Ejemplo de interacción con el asistente desde la aplicación web.

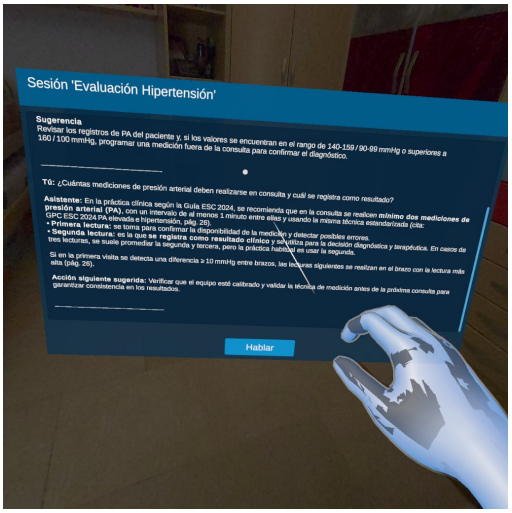
La segunda interfaz del prototipo corresponde a la aplicación de MR desarrollada para Meta Quest 3. Como se muestra en las Figuras 6.3 y 6.4, el asistente y el protocolo clínico se integran mediante paneles espaciales sobre el entorno físico, permitiendo realizar consultas, mantener el historial de la sesión y recorrer el protocolo mediante controladores o seguimiento de manos.



Figura 6.3: Vista general de la aplicación.



(a) Protocolo clínico y seguimiento de manos.



(b) Consulta al asistente e historial de la sesión.

Figura 6.4: Funcionamiento del prototipo de realidad mixta en Meta Quest 3.

6.2. Metodología de evaluación

Para realizar la evaluación se utilizó como documento de referencia la *Guía ESC 2024 sobre el manejo de la presión arterial elevada y la hipertensión* [McEvoy et al., 2024]. Este documento se seleccionó por su relevancia clínica, su extensión suficiente para exigir recuperación documental sobre múltiples apartados y su adecuación al caso de uso planteado en este trabajo.

Con el objetivo de evitar interferencias procedentes de otras fuentes documentales, se creó una sesión específica de evaluación que contenía únicamente esta guía. El recuperador se configuró para devolver los cinco fragmentos considerados más próximos semánticamente a cada consulta (`top_k = 5`).

Se construyó un conjunto de 20 preguntas cuya respuesta se encontraba en la documentación proporcionada. Dicho conjunto cubre distintos tipos de consulta, incluyendo definiciones, recomendaciones terapéuticas, objetivos clínicos y aspectos procedimentales de la guía. El listado completo de preguntas puede consultarse en el Anexo B.1.

Adicionalmente, se incluyeron cinco preguntas deliberadamente ajenas al contenido de la guía. Estas consultas se recogen en el Anexo B.2. Su finalidad fue analizar el comporta-

miento del sistema cuando la documentación disponible no contenía evidencia suficiente para responder. La evaluación se dividió en tres niveles:

- Evaluación de la recuperación documental, considerando únicamente los fragmentos recuperados por el sistema.
- Evaluación extremo a extremo, incluyendo recuperación, generación de la respuesta y devolución de las fuentes utilizadas.
- Evaluación del comportamiento frente a preguntas para las que no existía evidencia en la documentación.

Para evaluar la recuperación se utilizaron las métricas Hit@1, Hit@3, Hit@5, Mean Reciprocal Rank (MRR) y Precision@5.

La métrica Hit@K indica el porcentaje de consultas para las que aparece al menos un fragmento relevante entre los primeros K resultados recuperados. El MRR valora además la posición en la que aparece por primera vez dicho fragmento, otorgando mayor puntuación cuanto más arriba se encuentre en el ranking. Por su parte, Precision@5 representa la proporción media de fragmentos relevantes dentro de los cinco recuperados.

En la evaluación final, la relevancia se determinó en función del contenido presente en cada fragmento, en lugar de utilizar exclusivamente las páginas físicas del documento. Esta modificación permitió evitar penalizaciones incorrectas en aquellos casos en los que una misma recomendación aparecía repetida en diferentes secciones de la guía.

Como parte de la preparación de la evaluación, se compararon también dos modelos para la representación vectorial de los textos. Durante las primeras etapas del desarrollo se utilizó el modelo *bge-large-en* [Xiao et al., 2023] para la representación vectorial de los textos. Sin embargo, debido a que tanto la documentación utilizada como las consultas se encuentran principalmente en español, se evaluó también el modelo `intfloat/multilingual-e5-large` denominado en adelante *multilingual-e5-large* [Wang et al., 2024].

Ambos modelos fueron comparados utilizando los mismos 286 fragmentos documentales y las mismas 20 consultas. En el caso de *multilingual-e5-large*, las consultas se codificaron utilizando el prefijo `query:`, mientras que los fragmentos documentales emplearon el prefijo `passage:`

Los resultados obtenidos se muestran en la Tabla 6.1.

La mejora obtenida mediante *multilingual-e5-large* fue especialmente significativa en

Tabla 6.1: Comparación de los modelos de embeddings evaluados.

Métrica	BGE	Multilingual-E5
Hit@1	35,0 %	55,0 %
Hit@3	70,0 %	90,0 %
Hit@5	70,0 %	95,0 %
MRR	0,518	0,713
Posición media de la primera evidencia relevante	4,10	2,40

Hit@5, que pasó del 70,0 % al 95,0 %. También se produjo una mejora considerable del MRR y de la posición media de la primera evidencia relevante.

A partir de estos resultados se decidió utilizar *multilingual-e5-large* como modelo de embeddings en la versión final del sistema y volver a indexar la documentación utilizada durante la evaluación.

6.3. Evaluaciones realizadas

A partir de la metodología descrita anteriormente, se realizaron distintas pruebas destinadas a analizar por separado la recuperación documental, el comportamiento extremo a extremo del asistente, su respuesta ante consultas sin evidencia disponible y el rendimiento temporal del sistema.

6.3.1. Recuperación de información

Una vez sustituido el modelo de embeddings y reindexado el documento, se ejecutó el benchmark definitivo sobre las 20 preguntas contestables.

Los resultados obtenidos se presentan en la Tabla 6.2.

El Hit@5 del 95 % implica que una de las veinte consultas no recuperó la evidencia necesaria entre los cinco primeros fragmentos. En el 90 % de los casos la evidencia apareció entre los tres primeros resultados, mientras que en el 55 % se situó directamente en primera posición.

El valor de MRR de 0,713 refleja que, de forma general, el primer fragmento relevante se encuentra en posiciones altas del ranking.

La Precision@5 del 47 % indica que, aunque habitualmente se recupera la evidencia

necesaria, el contexto incluye también fragmentos secundarios. En concreto, de los cinco fragmentos recuperados para cada consulta, aproximadamente 2,35 contienen de media la evidencia considerada directamente relevante. Esto no implica que el sistema responda correctamente únicamente al 47 % de las consultas, sino que existe margen de mejora en la reducción del ruido documental proporcionado posteriormente al modelo generativo.

Tabla 6.2: Resultados de la evaluación de la recuperación documental.

Métrica	**Resultado**	**Métrica**	**Resultado**
Hit@1	55,0 %	MRR	0,713
Hit@3	90,0 %	Precision@5	47,0 %
Hit@5	95,0 %	Latencia media	0,125 s
Latencia mediana	0,113 s	Latencia p95	0,156 s

La consulta P05, "¿Cuántas mediciones de presión arterial deben realizarse en consulta y cuál se registra como resultado?", no consiguió recuperar dentro del *top-5* un fragmento que contuviera toda la evidencia definida para la consulta. La evidencia de referencia utilizada para su evaluación incluía el número de mediciones necesarias, el intervalo entre ellas y el cálculo del valor final.

Este caso resulta especialmente relevante, ya que la información sí estaba presente en el documento original, pero el recuperador no consiguió situarla entre los cinco primeros fragmentos. Por tanto, el fallo se produce en la etapa de recuperación y no en la generación de la respuesta.

6.3.2. Respuestas generadas

Tras evaluar de forma independiente el recuperador, se analizó el comportamiento completo del sistema. Para cada una de las 20 preguntas se realizó una petición al endpoint de conversación utilizando el mismo flujo empleado por la aplicación real.

El resultado incluyó tanto la recuperación de los fragmentos relevantes como la construcción del contexto, la llamada al modelo de lenguaje y la generación final de la respuesta.

Los resultados generales se muestran en la Tabla 6.3.

El sistema consiguió generar una respuesta final visible en el 90 % de las consultas. En dos de las veinte preguntas el modelo de lenguaje no produjo contenido final válido

Tabla 6.3: Resultados de la evaluación extremo a extremo.

Métrica	Resultado	Métrica	Resultado
Preguntas evaluadas	20	Respuestas completadas	18
Fallos de generación	2	Tasa de finalización	90,0 %
Fuente relevante presente	95,0 %	Latencia media	14,56 s
Latencia mediana	14,15 s	Latencia p95	24,62 s
Latencia mínima	6,87 s	Latencia máxima	24,64 s

incluso después de realizar un segundo intento de generación. En estos casos, la aplicación devolvió el mensaje de seguridad definido como mecanismo de *fallback*, evitando que el usuario recibiera una respuesta vacía.

Los dos fallos se produjeron en las preguntas P10 y P13. En ambos casos, el recuperador había proporcionado evidencia relevante entre los fragmentos recuperados, por lo que el fallo se produjo durante la etapa de generación y no durante la recuperación documental.

Durante las pruebas se observó que el modelo podía producir contenido interno de razonamiento sin llegar a generar contenido visible en el campo de respuesta. Debido a ello se implementó un mecanismo consistente en realizar un segundo intento de generación cuando la primera salida no contenía una respuesta final suficiente. Si ambos intentos fallaban, se devolvía un mensaje controlado al usuario.

Este comportamiento permite diferenciar dos tipos de fallo dentro de la arquitectura: los errores debidos a que la información necesaria no es recuperada y los errores que aparecen posteriormente durante la generación, incluso cuando el contexto proporcionado al modelo contiene la respuesta correcta.

La presencia de una fuente relevante entre las fuentes devueltas por el sistema alcanzó el 95 %, coherente con el Hit@5 obtenido durante la evaluación del recuperador.

6.3.3. Comportamiento ante ausencia de evidencia

Para estudiar el comportamiento del sistema cuando la documentación disponible no contiene la información solicitada, se utilizaron cinco consultas deliberadamente ajenas al ámbito de la guía indexada. Estas consultas se recogen en el Anexo B.2.

El objetivo de esta prueba no consistía en evaluar el conocimiento general del modelo de lenguaje, sino en comprobar si el sistema indicaba explícitamente que la documentación

disponible era insuficiente para proporcionar una respuesta fundamentada.

La detección automática de abstención se realizó mediante un conjunto de patrones textuales asociados a expresiones como la ausencia de información, la falta de cobertura del documento o la imposibilidad de responder a partir de la documentación proporcionada.

En la ejecución final, este mecanismo detectó una abstención explícita en una de las cinco consultas, correspondiente a una tasa del 20 %. Este resultado pone de manifiesto que el control mediante instrucciones de *prompt* no garantiza que el modelo limite siempre su respuesta a la documentación recuperada.

La métrica debe interpretarse además con cautela, ya que se basa en patrones textuales y puede no reconocer todas las formulaciones posibles de abstención. No obstante, los resultados muestran la necesidad de reforzar en trabajos futuros los mecanismos destinados a impedir la generación de información externa cuando no existe evidencia suficiente en las fuentes asociadas a la sesión.

6.3.4. Rendimiento temporal

El análisis temporal permitió diferenciar el coste asociado a la recuperación documental del correspondiente al proceso completo de generación de una respuesta. La Tabla 6.4 resume los principales tiempos registrados durante las pruebas.

Tabla 6.4: Resumen de los tiempos obtenidos durante la evaluación.

Operación	Media	p95
Recuperación vectorial	0,125 s	0,156 s
Consulta extremo a extremo	14,56 s	24,62 s

Los resultados muestran una diferencia clara entre ambas operaciones. La recuperación vectorial presentó una latencia media de 0,125 segundos y un percentil 95 de 0,156 segundos, mientras que una consulta extremo a extremo requirió de media 14,56 segundos y alcanzó un percentil 95 de 24,62 segundos. Por tanto, la búsqueda de información en el índice vectorial representa una parte reducida del tiempo total de respuesta, concentrándose la mayor parte de la latencia en las etapas posteriores de construcción del contexto y, principalmente, en la generación mediante el modelo de lenguaje alojado en el servidor externo.

Esta diferencia resulta relevante desde el punto de vista de la experiencia de usuario.

En la aplicación web, la generación progresiva permite mostrar el contenido conforme se recibe, reduciendo la percepción de espera aunque no disminuya el tiempo total necesario para completar la respuesta. La aplicación de MR, en cambio, espera actualmente a recibir la respuesta completa antes de mostrarla, por lo que este tiempo resulta más perceptible para el usuario.

6.4. Análisis de costes

Para estimar el coste del proyecto se han considerado la dedicación al desarrollo y los principales recursos hardware utilizados. La dedicación total se estima en aproximadamente 200 horas, incluyendo análisis, diseño, implementación, pruebas y documentación.

Para estimar el coste de personal se ha tomado como referencia el XIX Convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública. [Boletín Oficial del Estado, 2025, Boletín Oficial del Estado, 2026]. Considerando una jornada anual de 1.800 horas, el coste salarial bruto se estima en:

$$\frac{29.079{,}99}{1.800} \approx 16{,}16\ €/\text{h}.$$

En cuanto al hardware, el equipo portátil utilizado durante el desarrollo fue adquirido inicialmente para la realización del máster por un importe aproximado de 780 €. Por otro lado, tanto el dispositivo Meta Quest 3 utilizado para el desarrollo y las pruebas de la aplicación de MR como el servidor empleado para ejecutar el modelo de lenguaje fueron proporcionados por la UCLM. La Tabla 6.5 resume la estimación realizada.

Tabla 6.5: Estimación de los costes asociados al desarrollo del proyecto.

Concepto	**Criterio de cálculo**	**Coste**
Personal	200 h × 16,16 €/h + 33,35 % de costes empresariales	4.309,87 €
Equipo portátil	Coste aproximado de adquisición	780,00 €
Meta Quest 3	Recurso proporcionado por la UCLM	0,00 €
Servidor de inferencia	Infraestructura proporcionada por la UCLM	0,00 €
Total estimado		**5.089,87 €**

7. Conclusiones y trabajo futuro

En este capítulo se analiza el grado de cumplimiento de los objetivos planteados, se presentan las principales conclusiones obtenidas tras el desarrollo y evaluación del sistema, se justifican las competencias abordadas y se recogen las principales líneas de trabajo futuro.

7.1. Cumplimiento de los objetivos

El objetivo general del trabajo puede considerarse alcanzado, ya que se ha diseñado e implementado un prototipo funcional de asistente virtual contextual y proactivo basado en LLMs, capaz de consultar documentación clínica mediante lenguaje natural, mantener contexto asociado a las sesiones y ofrecer sus funcionalidades tanto desde una interfaz convencional como desde una aplicación de MR para Meta Quest 3.

En relación con los objetivos específicos:

- **OE1. Estudio de arquitecturas y tecnologías para la integración de LLMs:** cumplido mediante el análisis de modelos de lenguaje, recuperación semántica, sistemas RAG y herramientas de integración aplicables al ámbito sanitario.
- **OE2. Diseño del asistente virtual contextual:** cumplido mediante el desarrollo de un asistente capaz de responder consultas en lenguaje natural utilizando documentación clínica y manteniendo trazabilidad de las fuentes empleadas.

- **OE3. Desarrollo del backend del asistente**: cumplido mediante la implementación de los servicios necesarios para gestionar sesiones, conversaciones, documentos y tareas, así como coordinar la recuperación y generación de respuestas.
- **OE4. Desarrollo de mecanismos de interacción y gestión**: cumplido mediante una aplicación web para gestionar sesiones, documentos, tareas, consultas y fuentes asociadas a las respuestas.
- **OE5. Incorporación de comportamiento contextual y proactivo**: cumplido mediante persistencia del contexto conversacional, generación de preguntas sugeridas, resúmenes y extracción automática de tareas.
- **OE6. Definición e integración de mecanismos de interacción multimodal**: cumplido mediante interacción por voz, controladores, seguimiento de manos y visualización espacial dentro de la aplicación de MR.
- **OE7. Desarrollo e integración de la aplicación de MR**: cumplido mediante una aplicación para Meta Quest 3 con *passthrough*, paneles espaciales manipulables y navegación de protocolos clínicos.
- **OE8. Evaluación técnica y funcional del prototipo**: parcialmente cumplido, ya que se realizaron pruebas técnicas y funcionales, pero no se completó la evaluación de usabilidad con profesionales sanitarios.

7.2. Conclusiones generales

El desarrollo realizado demuestra la viabilidad técnica de integrar LLMs, recuperación documental y MR dentro de una misma solución orientada al apoyo del personal de atención primaria. La arquitectura desarrollada permite mantener contexto entre interacciones, recuperar información asociada a una sesión y proporcionar acceso al asistente tanto desde una aplicación web como desde Meta Quest 3.

La evaluación muestra que la recuperación documental ofrece resultados satisfactorios, aunque el comportamiento del modelo generativo continúa presentando limitaciones, especialmente cuando la documentación disponible no contiene evidencia suficiente. Por ello, la trazabilidad de las fuentes y la supervisión del profesional resultan elementos esenciales en un sistema de estas características.

En conjunto, el prototipo constituye una base funcional sobre la que continuar la integración del asistente dentro del proyecto MRP-5G, sin que los resultados obtenidos deban

interpretarse como una validación clínica ni como soporte para la toma autónoma de decisiones.

Los resultados obtenidos también muestran que el comportamiento final del asistente depende de forma conjunta de las etapas de recuperación y generación. Una recuperación adecuada no garantiza por sí sola una respuesta correcta, del mismo modo que una arquitectura RAG no evita completamente la generación de información no respaldada por las fuentes disponibles. Esto refuerza la necesidad de mantener mecanismos de trazabilidad, supervisión humana y validación adicional antes de plantear un uso en escenarios clínicos reales.

7.3. Competencias adquiridas

El desarrollo del presente TFM ha permitido abordar las competencias CP01 y CP02 establecidas para el trabajo.

- **CP01. Aplicar conceptos, técnicas y métodos avanzados de Inteligencia Artificial en entornos que requieran representación del conocimiento y razonamiento, aprendizaje automático, ayuda a la toma de decisiones, percepción e interacción con el entorno, u otros.**

  Esta competencia se ha abordado mediante la aplicación de diferentes técnicas de IA en el diseño del asistente. La representación del conocimiento documental se realiza mediante *embeddings* y recuperación semántica, mientras que la arquitectura RAG permite incorporar dicho conocimiento al contexto utilizado por el LLM y conservar la trazabilidad de las fuentes. Asimismo, el sistema integra mecanismos de interacción mediante lenguaje natural, voz y MR, así como la navegación de protocolos clínicos, combinando técnicas de IA con mecanismos de interacción orientados al apoyo del profesional sanitario.

- **CP02. Planificar y ejecutar proyectos de desarrollo y despliegue de soluciones a problemas complejos que requieran de técnicas y métodos de Inteligencia Artificial en entornos que impliquen conducta inteligente, aprendizaje e interacción, conforme a procesos de ingeniería de proyectos.**

  Esta competencia se ha abordado mediante la planificación, diseño, implementación, integración y evaluación de una solución completa de IA formada por un *backend*,

una aplicación web y una aplicación de MR. El desarrollo se ha organizado mediante un enfoque iterativo e incremental, integrando progresivamente persistencia, recuperación documental, generación mediante LLM e interacción multimodal. Además, se han utilizado mecanismos de control de versiones, contenerización y despliegue mediante Git y Docker, junto con pruebas funcionales y de rendimiento para validar los principales componentes del sistema.

7.4. Líneas de trabajo futuro

La principal línea de continuación consiste en validar el sistema con profesionales sanitarios. Esta evaluación permitiría analizar aspectos que no pueden medirse únicamente mediante pruebas técnicas, como la usabilidad percibida, la utilidad práctica de las respuestas, la carga cognitiva asociada a la interacción y la adecuación de la aplicación de MR al contexto clínico.

También sería conveniente ampliar la evaluación utilizando un conjunto mayor de documentos y consultas, incluyendo múltiples fuentes dentro de una misma sesión y posibles contradicciones entre ellas. En este contexto podrían estudiarse técnicas como el *reranking*, la recuperación híbrida o la selección dinámica del número de fragmentos, con el objetivo de reducir el ruido proporcionado al modelo y mejorar la precisión del proceso de recuperación.

En la aplicación de MR, futuras versiones podrían permitir seleccionar directamente la sesión activa, reconstruir visualmente su historial y adaptar de forma dinámica el protocolo mostrado al contexto de trabajo. Asimismo, la incorporación de síntesis de voz permitiría complementar la interacción actual y reducir la necesidad de consultar continuamente los paneles virtuales.

Finalmente, una evolución hacia escenarios de uso más próximos a la práctica real requeriría profundizar en aspectos relacionados con seguridad, privacidad, interoperabilidad clínica y mecanismos adicionales de control y validación de las respuestas generadas.

Referencias bibliográficas

[Amugongo et al., 2025] Amugongo, L. M., Mascheroni, P., Brooks, S., Doering, S., and Seidel, J. (2025). Retrieval augmented generation for large language models in healthcare: A systematic review. *PLOS Digital Health*, 4(6):e0000877. Disponible en: https://doi.org/10.1371/journal.pdig.0000877.

[Bedi et al., 2025] Bedi, S., Liu, Y., Orr-Ewing, L., Dash, D., Koyejo, S., Callahan, A., Fries, J. A., Wornow, M., Swaminathan, A., Lehmann, L. S., et al. (2025). Testing and evaluation of health care applications of large language models: A systematic review. *JAMA*, 333(4):319–328. Disponible en: https://doi.org/10.1001/jama.2024.21700.

[Billinghurst et al., 2015] Billinghurst, M., Clark, A., and Lee, G. (2015). A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8(2-3):73–272. Disponible en: https://doi.org/10.1561/1100000049.

[Boletín Oficial del Estado, 2025] Boletín Oficial del Estado (2025). Xix convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública. BOE-A-2025-7766. Disponible en: https://www.boe.es/eli/es/res/2025/04/04/(10).

[Boletín Oficial del Estado, 2026] Boletín Oficial del Estado (2026). Actualización de las tablas salariales del xix convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública. BOE-A-2026-9024. Disponible en: https://www.boe.es/eli/es/res/2026/04/13/(3).

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

[del Hoyo Abad, 2024] del Hoyo Abad, P. (2024). Intelligent system based on mixed reality for diagnosis and assistance in primary care. Trabajo Fin de Grado, Universidad de Castilla-La Mancha.

[Duggan et al., 2025] Duggan, M. J., Gervase, J., Schoenbaum, A., Hanson, W., Howell, J. T., Sheinberg, M., and Johnson, K. B. (2025). Clinician experiences with ambient scribe technology to assist with documentation burden and efficiency. *JAMA Network Open*, 8(2):e2460637. Disponible en: https://doi.org/10.1001/jamanetworkopen.2024.60637.

[European Union, 2016] European Union (2016). Regulation (eu) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Union. General Data Protection Regulation. Disponible en: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[European Union, 2024] European Union (2024). Regulation (eu) 2024/1689 laying down harmonised rules on artificial intelligence. Official Journal of the European Union. Artificial Intelligence Act. Disponible en: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

[García et al., 2025a] García, F. M., Essalhi, M., Espejo, S., Sánchez-Sobrino, S., Albusac, J. A., and Vallejo, D. (2025a). Mixed reality-based platform for remote support and diagnosis in primary care: A position paper. In *Proceedings of the 27th International Conference on Enterprise Information Systems*, volume 2, pages 565–572. SciTePress. Disponible en: https://doi.org/10.5220/0013352100003929.

[García et al., 2025b] García, F. M., Schez-Sobrino, S., Glez-Morcillo, C., Castro-Schez, J. J., Albusac, J. A., and Vallejo, D. (2025b). Rtc-mr: A webrtc-based framework for real-time communication in mixed reality. *Software Impacts*, 23:100727.

[Irving et al., 2017] Irving, G., Neves, A. L., Dambha-Miller, H., Oishi, A., Tagashira, H., Verho, A., and Holden, J. (2017). International variations in primary care physician consultation time: a systematic review of 67 countries. *BMJ Open*, 7(10):e017902. Disponible en: https://doi.org/10.1136/bmjopen-2017-017902.

[Ji et al., 2023] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38. Disponible en: https://doi.org/10.1145/3571730.

[Larman and Basili, 2003] Larman, C. and Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6):47–56. Disponible en: https://doi.org/10.1109/MC.2003.1204375.

[Lewis et al., 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

[Martin et al., 2020] Martin, G., Koizia, L., Kooner, A., et al. (2020). Use of the HoloLens2 mixed reality headset for protecting health care workers during the COVID-19 pandemic. *Journal of Medical Internet Research*, 22(8):e21486.

[McEvoy et al., 2024] McEvoy, J. W., McCarthy, C. P., Bruno, R. M., Brouwers, S., Canavan, M. D., Ceconi, C., Christodorescu, R. M., Daskalopoulou, S. S., Ferro, C. J., Gerdts, E., Hanssen, H., Harris, J., Lauder, L., McManus, R. J., Molloy, G. J., Rahimi, K., Regitz-Zagrosek, V., Rossi, G. P., Sandset, E. C., Scheenaerts, B., Staessen, J. A., Uchmanowicz, I., Volterrani, M., Touyz, R. M., et al. (2024).

2024 esc guidelines for the management of elevated blood pressure and hypertension. *European Heart Journal*, 45(38):3912–4018. Disponible en: https://doi.org/10.1093/eurheartj/ehae178.

[Microsoft, 2026] Microsoft (2026). Microsoft dragon copilot. Microsoft for Healthcare. Consultado en septiembre de 2026. Disponible en: https://www.microsoft.com/en-us/health-solutions/clinical-workflow/dragon-copilot.

[Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, E77-D(12):1321–1329.

[Peleg, 2013] Peleg, M. (2013). Computer-interpretable clinical guidelines: A methodological review. *Journal of Biomedical Informatics*, 46(4):744–763. Disponible en: https://doi.org/10.1016/j.jbi.2013.06.009.

[Schwaber and Sutherland, 2020] Schwaber, K. and Sutherland, J. (2020). The scrum guide: The definitive guide to scrum. Disponible en: https://scrumguides.org/scrum-guide.html.

[Singhal et al., 2023] Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. (2023). Large language models encode clinical knowledge. *Nature*, 620:172–180. Disponible en: https://doi.org/10.1038/s41586-023-06291-2.

[Topol, 2019] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25:44–56. Disponible en: https://doi.org/10.1038/s41591-018-0300-7.

[Tu et al., 2025] Tu, T., Schaekermann, M., Palepu, A., Saab, K., Freyberg, J., Tanno, R., Wang, A., Li, B., Amin, M., Cheng, Y., et al. (2025). Towards conversational diagnostic artificial intelligence. *Nature*, 642:442–450. Disponible en: https://doi.org/10.1038/s41586-025-08866-7.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

[Wang et al., 2024] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Multilingual e5 text embeddings: A technical report. Disponible en: https://arxiv.org/abs/2402.05672.

[WHO and UNICEF, 2020] WHO and UNICEF (2020). *Operational Framework for Primary Health Care: Transforming Vision into Action*. World Health Organization, Geneva. Disponible en: https://www.who.int/publications/i/item/9789240017832.

[Worlikar et al., 2023] Worlikar, H., Coleman, S., Kelly, J., O'Connor, S., Murray, A., McVeigh, T., Doran, J., McCabe, I., and O'Keeffe, D. (2023). Mixed reality platforms in telehealth delivery: Scoping review. *JMIR Biomedical Engineering*, 8:e42709. Disponible en: https://doi.org/10.2196/42709.

[Xiao et al., 2023] Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., and Nie, J.-Y. (2023). C-pack: Packed resources for general chinese embeddings. Disponible en: https://arxiv.org/abs/2309.07597.

A. Guía de instalación y ejecución

En este anexo se describen los pasos necesarios para obtener, configurar y ejecutar los principales componentes del prototipo desarrollado. La solución está compuesta por una aplicación web, un *backend*, los servicios de persistencia PostgreSQL y Qdrant, y una aplicación de realidad mixta desarrollada en Unity para Meta Quest 3.

El modelo de lenguaje utilizado por el sistema no forma parte del despliegue local, ya que se ejecuta de forma independiente en infraestructura de la UCLM. Por tanto, para disponer de todas las funcionalidades es necesario contar con acceso al servicio de inferencia correspondiente.

A.1. Repositorio del proyecto

El código fuente del proyecto se encuentra disponible públicamente en GitHub:

https://github.com/chemabr24/tfm-llm-assistant-mrp5g

El repositorio contiene tres componentes principales:

- `backend/`: servicio desarrollado con FastAPI que implementa la lógica del asistente, la recuperación documental, la persistencia, la transcripción de voz y la comunicación con el modelo de lenguaje.
- `frontend/`: aplicación web desarrollada con Angular.
- `mr/`: proyecto de Unity correspondiente a la aplicación de realidad mixta para Meta Quest 3.

El repositorio puede obtenerse mediante Git ejecutando:

```
git clone https://github.com/chemabr24/tfm-llm-assistant-mrp5g.git
cd tfm-llm-assistant-mrp5g
```

A.2. Requisitos previos

Para ejecutar la aplicación web y el *backend* mediante contenedores se requiere tener instalados Git y Docker con soporte para Docker Compose.

Además, el sistema necesita acceso al servicio externo que aloja el modelo de lenguaje, junto con las credenciales correspondientes. Las funcionalidades de autenticación mediante *magic links* y recordatorios requieren también disponer de una cuenta de correo configurada para el envío mediante SMTP.

Durante la primera ejecución del *backend* pueden descargarse los modelos utilizados para la generación de *embeddings* y la transcripción mediante Whisper, por lo que es necesario disponer de conexión a Internet.

Para ejecutar o modificar la aplicación de realidad mixta se requiere un dispositivo Meta Quest 3 y Unity 2022.3.62f3, que puede instalarse mediante Unity Hub. Durante la instalación de Unity deben incluirse los módulos *Android Build Support*, *Android SDK & NDK Tools* y *OpenJDK*, necesarios para generar una compilación destinada al visor.

Las dependencias del proyecto definidas en `mr/Packages/manifest.json` son restauradas automáticamente por Unity al abrir el proyecto. Entre las principales dependencias utilizadas se encuentran Meta XR All-in-One SDK 201.0.0, XR Interaction Toolkit 2.6.5, XR Plugin Management 4.7.0 y Oculus XR Plugin 4.5.5.

Para ejecutar el proyecto directamente desde el editor mediante Meta Quest Link es necesario disponer también de la aplicación Meta Quest Link en el equipo. El visor debe tener habilitado el modo desarrollador y autorizar la depuración USB cuando sea necesario.

A.3. Configuración del entorno

Antes de iniciar los contenedores debe crearse el fichero de configuración del *backend* a partir de la plantilla incluida en el repositorio:

```
cp backend/.env.example backend/.env
```

En sistemas Windows puede realizarse la misma operación copiando manualmente `backend/.env.example` y renombrando la copia como `.env`.

Las principales variables que deben configurarse son las siguientes:

```
LLM_API_URL=http://<host_llm>:<puerto>
LLM_API_KEY=<api_key>
LLM_MODEL=gpt-oss:20b

EMBEDDING_MODEL=intfloat/multilingual-e5-large

MAIL_USERNAME=<email>
MAIL_PASSWORD=<application_password>
MAIL_FROM=<email>
MAIL_SERVER=smtp.gmail.com
MAIL_PORT=587
```

Las variables correspondientes a PostgreSQL y Qdrant son sobrescritas automáticamente por la configuración de Docker Compose para utilizar los servicios desplegados en la misma red de contenedores.

Las credenciales reales no deben incorporarse al repositorio. El fichero `.env.example` se utiliza únicamente como plantilla de referencia.

A.4. Despliegue mediante Docker Compose

Una vez configurado el fichero `backend/.env`, los componentes principales pueden construirse e iniciarse desde el directorio raíz del proyecto mediante:

```
docker compose up --build
```

La primera construcción de las imágenes puede requerir varios minutos, especialmente durante la instalación de las dependencias utilizadas para la generación de *embeddings* y la transcripción de audio.

La configuración inicia cuatro servicios:

- PostgreSQL, encargado de la persistencia relacional.
- Qdrant, utilizado para el almacenamiento y recuperación vectorial.
- el *backend* desarrollado con FastAPI.
- el *frontend* Angular servido mediante Nginx.

Una vez finalizado el arranque, la aplicación web queda accesible desde:

```
http://localhost
```

El *backend* expone además un punto de comprobación de estado en:

```
http://localhost:8000/health
```

La documentación interactiva de la API puede consultarse en:

```
http://localhost:8000/docs
```

Las tablas necesarias en PostgreSQL se crean automáticamente durante el arranque del *backend*. PostgreSQL y Qdrant utilizan volúmenes persistentes, por lo que los datos se conservan aunque los contenedores se detengan o se reconstruyan.

La incorporación inicial de documentos puede requerir varios minutos, ya que durante la misma petición se realiza la extracción del texto, la división en fragmentos, la generación de *embeddings* y su almacenamiento en Qdrant. Durante este proceso, la aplicación web muestra el documento como en procesamiento hasta que finaliza la indexación.

Para detener la ejecución puede utilizarse:

```
docker compose down
```

Si además se desea eliminar los volúmenes persistentes y reiniciar completamente los datos almacenados:

```
docker compose down -v
```

Este último comando elimina de forma permanente los datos almacenados en PostgreSQL y Qdrant y debe utilizarse únicamente cuando se desee reiniciar completamente el entorno.

A.5. Ejecución de la aplicación de realidad mixta

El proyecto puede incorporarse a Unity Hub seleccionando el directorio `mr/`. Al abrirlo por primera vez, Unity restaura las dependencias declaradas en el proyecto. Una vez finalizado este proceso, debe abrirse la escena:

```
Assets/Scenes/MainScene.unity
```

Antes de ejecutar la aplicación debe configurarse la dirección del *backend* accesible desde el dispositivo. Esta dirección se encuentra en el componente `APIManager`, mediante el parámetro `apiBaseUrl`, y debe seguir el formato:

```
http://<IP_DEL_EQUIPO>:8000/api
```

El equipo que ejecuta el *backend* y Meta Quest 3 deben poder comunicarse a través de la misma red. La dirección `localhost` no debe utilizarse en la aplicación de realidad mixta, ya que desde el visor haría referencia al propio dispositivo y no al equipo donde se está ejecutando FastAPI.

Una vez configurada la dirección del servicio, el prototipo puede ejecutarse directamente desde el editor mediante el botón *Play* con Meta Quest Link activo o, alternativamente, generarse como aplicación Android e instalarse en Meta Quest 3 para su ejecución autónoma.

La escena principal integra la consulta al asistente, la entrada mediante voz, la visualización del historial, la interacción con paneles espaciales y la navegación del protocolo clínico mediante controladores y seguimiento de manos.

B. Preguntas utilizadas en la evaluación

Este anexo recoge las preguntas utilizadas para evaluar el comportamiento del sistema de recuperación documental y del asistente extremo a extremo. El conjunto incluye consultas cuya respuesta se encuentra respaldada por la documentación indexada y consultas para las que no existe evidencia suficiente, permitiendo evaluar tanto la capacidad de recuperación como el comportamiento del sistema ante ausencia de contexto documental relevante.

Las preguntas se identifican mediante un código para facilitar su referencia durante el análisis de los resultados. El conjunto de evaluación está compuesto por un total de 25 consultas, distribuidas como se muestra en la Tabla B.1.

Tabla B.1: Composición del conjunto de preguntas utilizado en la evaluación.

Tipo de consulta	Identificadores	Número
Con evidencia documental	P01–P20	20
Sin evidencia documental	S01–S05	5
Total		**25**

B.1. Preguntas con evidencia documental

Estas preguntas fueron seleccionadas de forma que la documentación utilizada durante la evaluación contuviera información relevante para responderlas. El conjunto incluye

cuestiones sobre definición y diagnóstico de la hipertensión, medición de la presión arterial, estimación del riesgo cardiovascular, tratamiento, seguimiento y situaciones clínicas específicas. Su utilización permite comprobar si el sistema recupera fragmentos adecuados y los emplea correctamente durante la generación de la respuesta.

Tabla B.2: Preguntas de evaluación con evidencia documental.

ID	Pregunta	Aspecto evaluado
P01	¿A partir de qué valores de presión arterial en consulta se define hipertensión según la guía ESC 2024?	Definición
P02	¿Qué intervalo define la nueva categoría de presión arterial elevada en consulta?	Clasificación
P03	¿Cuál es el umbral de hipertensión cuando la presión arterial se mide en el domicilio mediante AMPA?	Medición domiciliaria
P04	¿Qué valores de MAPA de 24 horas confirman hipertensión?	Medición ambulatoria
P05	¿Cuántas mediciones de presión arterial deben realizarse en consulta y cuál se registra como resultado?	Técnica de medición
P06	¿Cómo define la guía la hipotensión ortostática?	Definición
P07	En un paciente con 145/95 mmHg en consulta, ¿cómo recomienda la guía confirmar el diagnóstico de hipertensión?	Confirmación diagnóstica
P08	¿Con qué frecuencia se debe considerar el cribado de presión arterial en adultos menores de 40 años y en mayores de 40?	Cribado
P09	¿Cuál es el objetivo recomendado de presión arterial sistólica para la mayoría de pacientes tratados, si se tolera bien?	Objetivo terapéutico
P10	¿Qué clases de fármacos se recomiendan como tratamientos antihipertensivos de primera línea?	Tratamiento
P11	¿Qué cantidad de ejercicio aeróbico semanal recomienda la guía para reducir la presión arterial y el riesgo cardiovascular?	Estilo de vida
P12	¿Qué recomienda la guía respecto al consumo de alcohol?	Estilo de vida

Tabla B.2 – continuación

ID	Pregunta	Aspecto evaluado
P13	Una vez controlada y estable la presión arterial con tratamiento, ¿con qué frecuencia mínima debe considerarse el seguimiento?	Seguimiento
P14	¿Qué umbral de SCORE2 o SCORE2-OP considera la guía como riesgo cardiovascular suficientemente alto en una persona con presión arterial elevada?	Riesgo cardiovascular
P15	¿Qué tratamiento recomienda la guía añadir en la hipertensión resistente no controlada pese al tratamiento de primera línea?	Hipertensión resistente
P16	¿Qué cifras de presión arterial pueden indicar una emergencia hipertensiva durante el embarazo?	Situación especial
P17	En pacientes con hipertensión y enfermedad renal crónica con TFGe superior a 20 ml/min/1,73 m^2, ¿qué grupo farmacológico recomienda la guía para mejorar resultados?	Enfermedad renal
P18	¿Qué diferencia de presión arterial sistólica entre ambos brazos se considera relevante y qué debe hacerse después?	Técnica de medición
P19	¿Cuánto tiempo debe permanecer el paciente en reposo antes de medir la presión arterial en consulta?	Técnica de medición
P20	¿Durante cuántos días recomienda la guía realizar la automedida de presión arterial en domicilio?	Medición domiciliaria

B.2. Preguntas sin evidencia documental

El segundo grupo está formado por consultas pertenecientes a ámbitos clínicos no cubiertos por la documentación indexada. Estas preguntas permiten comprobar si el asistente identifica la ausencia de evidencia suficiente y evita presentar como respaldada por la guía información que no se encuentra contenida en ella.

Tabla B.3: Preguntas de evaluación sin evidencia documental.

ID	Pregunta	Ámbito externo
S01	¿Qué pauta antibiótica recomienda esta guía para tratar una otitis media aguda?	Otorrino-laringología
S02	¿Cuál es la dosis de amoxicilina recomendada por esta guía para una neumonía adquirida en la comunidad?	Infección respiratoria
S03	¿Qué tratamiento recomienda esta guía para una fractura de fémur?	Traumatología
S04	¿Qué calendario vacunal infantil recomienda esta guía?	Pediatría
S05	¿Qué dosis de insulina rápida recomienda esta guía para una cetoacidosis diabética?	Diabetes