



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Máster en Data Science

Trabajo Fin de Máster

**Museion: Desarrollo de una Aplicación
Basada en IA Generativa para la
Interacción Personalizada en Entornos
Culturales (Museo Nacional
Thyssen-Bornemisza)**

Autor: Francisco Madrigal Puertas

Tutores: Concha Bielza Lozoya y Pedro Larrañaga Múgica

Madrid, Mayo - 2025

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster en Data Science

Título: Museion: Desarrollo de una Aplicación Basada en IA Generativa para la Interacción Personalizada en Entornos Culturales (Museo Nacional Thyssen-Bornemisza)

Mayo - 2025

Autor: Francisco Madrigal Puertas
Tutores: Concha Bielza Lozoya y Pedro Larrañaga Múgica
Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Índice general

1. Introducción	4
1.1. Motivación del proyecto	4
1.2. Contexto del proyecto	4
1.3. Fundamentos	5
1.3.1. <i>Large Language Models</i> (LLMs)	5
1.3.2. <i>Chatbot</i>	6
1.4. Objetivos	6
1.5. Contenidos del documento	7
2. Estado del arte y herramientas	9
2.1. Fundamentos de los LLMs	9
2.1.1. Enfoques	9
2.1.1.1. Preentrenamiento	9
2.1.1.2. Ajuste fino	10
2.1.1.3. <i>Reinforcement learning from human feedback</i> (RLHF) . . .	10
2.1.1.4. <i>Adapters</i>	11
2.1.1.5. <i>Prompt engineering</i>	11
2.1.1.6. <i>Retrieval augmented generation</i> (RAG)	11
2.1.1.7. Sistemas multiagentes	12
2.1.2. Modelos actuales	12
2.1.2.1. LLMs	12
2.1.2.2. Modelos de <i>embeddings</i>	13
2.2. Trabajos relacionados	14
2.2.1. Trabajos relacionados en entornos museísticos	14
2.2.1.1. Modelos de lenguaje como sistema de recomendación . .	14
2.2.1.2. <i>Chatbots</i> como herramientas de interacción en museos .	15
2.2.1.3. <i>Digital Curator Training Tool Box</i> (DCbox)	15
2.2.1.4. <i>VirtuWander</i>	15
2.2.1.5. GPTs aplicados a entornos museísticos	16
2.2.2. Trabajos relacionados enfocados en LLMs	16
2.2.2.1. Mecanismos de colaboración en soluciones multiagente .	16
2.2.2.2. Técnicas de RAG avanzadas	16
2.3. Herramientas	16
2.3.1. Python	17
2.3.2. Herramientas para <i>web scraping</i>	17
2.3.2.1. BeautifulSoup	18
2.3.2.2. Selenium	18
2.3.3. Inferidor - Servidor	18
2.3.4. SDKs	18
2.3.4.1. LlamaIndex	19

2.3.4.2. LangChain	19
2.3.5. Plataformas y servicios	20
2.3.5.1. Amazon Bedrock	20
2.3.5.2. Microsoft Azure	20
2.3.5.3. OpenAI API	20
2.3.5.4. Groq API	21
2.3.6. Bases de datos	21
2.3.6.1. MongoDB	21
2.3.6.2. Pinecone DB	21
2.3.7. <i>Streamlit</i>	21
3. Desarrollo	23
3.1. Obtención de los datos	23
3.2. Diseño de la aplicación	25
3.2.1. Sección “Información del museo”	25
3.2.2. Sección “Consultar obras”	26
3.2.3. Sección “Recorridos temáticos”	27
3.3. Arquitectura de datos	27
3.3.1. Base de datos no relacional	28
3.3.1.1. Colección de artistas	28
3.3.1.2. Colección de obras	29
3.3.1.3. Colección recorridos	30
3.3.1.4. Colección de tarifas	30
3.3.1.5. Colección de actividades	31
3.3.2. Base de datos vectorial	31
3.3.2.1. Índice de descripciones	32
3.3.2.2. Índice de localizaciones	33
3.4. Ingesta de datos	34
3.4.1. Ingesta en índice de descripciones	34
3.4.2. Ingesta en índice de localizaciones	34
3.5. Implementación de Museion	35
3.5.1. Gestión de la memoria	35
3.5.2. <i>Chatbot</i> de información del museo	36
3.5.3. <i>Chatbot</i> de información de obras	37
3.5.3.1. Agente enrutador	38
3.5.3.2. Agente localizador	39
3.5.3.3. Agente descriptor	40
3.5.3.4. Agente otras preguntas (<i>Other</i>)	42
3.5.3.5. Agente preguntas no relacionadas (<i>Non-related</i>)	43
3.5.4. <i>Chatbot</i> de recorridos temáticos	43
3.5.4.1. Selección de recorrido temático	44
3.5.4.2. Introducción al recorrido	44
3.5.4.3. Descripciones de la obra	44
3.5.4.4. Consultas sobre la obra	44
4. Demostración	46
4.1. <i>Landing page</i>	46
4.2. Sección de “Información del museo”	46
4.2.1. Inferencia del modelo	46
4.2.1.1. Pregunta 1	46
4.2.1.2. Pregunta 2	47
4.2.1.3. Pregunta 3	47

4.2.1.4. Pregunta 4	48
4.2.1.5. Pregunta 5	48
4.2.1.6. Pregunta 6	48
4.3. Sección de “Información de Obras”	49
4.3.1. Inferencia del modelo	49
4.3.1.1. Pregunta 1	49
4.3.1.2. Pregunta 2	49
4.3.1.3. Pregunta 3	49
4.3.1.4. Pregunta 4	50
4.3.1.5. Pregunta 5	50
4.3.1.6. Pregunta 6	51
4.3.1.7. Pregunta 7	51
4.3.1.8. Pregunta 8	51
4.4. Sección de “Recorridos temáticos”	52
4.4.1. Inferencia del modelo	53
4.4.1.1. Pregunta 1	53
4.4.1.2. Pregunta 2	54
4.4.1.3. Pregunta 3	56
5. Evaluación	57
5.1. Evaluación de la capacidad agencial del sistema multiagente de la sección “Consultar obras”	57
5.1.1. Método de evaluación utilizado	57
5.1.2. Datos utilizados	58
5.1.3. Resultados	58
5.1.3.1. Conclusiones de los resultados	61
5.2. Evaluación del agente conversacional de descripciones	62
5.2.1. Métricas de evaluación utilizadas	62
5.2.2. Metodología de evaluación	62
5.2.3. Datos utilizados	64
5.2.4. Resultados	65
6. Resultados y conclusiones	66
6.1. Resultados	66
6.2. Conclusiones personales	66
6.3. Impacto del proyecto	67
6.4. Trabajo futuro	68
Bibliografía	70
A. Apéndice de información incluida en Museion	73
A.1. Autores incluidos	73
A.2. Obras maestras incluidas	74
A.3. Recorridos temáticos incluidos	75

Resumen

Los avances en el campo de la inteligencia artificial generativa, particularmente en los modelos de lenguaje de gran escala (LLMs), han ocasionado una transformación significativa tanto en el ámbito social como cultural. Casos como el de ChatGPT han demostrado el vasto potencial de estas herramientas, que se han integrado progresivamente en las actividades cotidianas de la sociedad. Esta adopción no se limita únicamente a los usuarios individuales, sino que también ha sido impulsada por diversas instituciones, que están apostando por la incorporación de estas tecnologías en sus productos.

En este contexto, se plantea la adaptación de estas tecnologías al ámbito cultural, centrándose en el desarrollo de un asistente virtual para el Museo Nacional Thyssen-Bornemisza. Este asistente tiene como objetivo guiar a los visitantes durante su recorrido por el museo y responder a sus consultas sobre las obras y los autores expuestos, así como abordar otras inquietudes relacionadas con su visita.

El propósito principal de este proyecto es integrar la tecnología en los entornos museísticos con el fin de promover la innovación dentro del sector cultural. Se busca, además, crear experiencias personalizadas para los visitantes, acercar la cultura a las nuevas generaciones nativas digitales, y ofrecer nuevas formas de interacción durante las visitas, haciendo más accesible y atractiva la experiencia museística para toda la población.

A lo largo de este Trabajo Fin de Máster se detallarán los aspectos técnicos y teóricos que sustentan la implementación de esta herramienta, basada en ingeniería de inteligencia artificial.

Palabras Clave: IA generativa, Ingeniería de IA, LLMs, Asistente virtual, Museos

Abstract

Advancements in the field of generative artificial intelligence, particularly in large language models (LLMs), have led to a significant transformation in both social and cultural spheres. Cases such as ChatGPT have demonstrated the vast potential of these tools, which have been progressively integrated into the daily activities of society. This adoption is not limited to individual users, but has also been embraced by numerous institutions that are betting on incorporating these technologies into their products.

In this context, the adaptation of these technologies to the cultural domain is proposed, focusing on the development of a virtual assistant for the Museo Nacional Thyssen-Bornemisza. This assistant aims to guide visitors through the museum and respond to their queries about the exhibited works and artists, as well as address other concerns related to their visit.

The primary goal of this project is to integrate technology into museum environments to foster innovation within the cultural sector. Additionally, it seeks to create personalized experiences for visitors, bring culture closer to new, digitally native generations, and offer new ways to engage during museum visits, making the museum experience more accessible and attractive to the general public.

Throughout this Master's Thesis, the technical and theoretical aspects underlying the implementation of this tool, based on artificial intelligence engineering, will be detailed.

Keywords: Generative AI, AI engineering, LLMs, Virtual assistant, Museums

Agradecimientos

A mis tutores, Pedro y Concha, por proponer y guiar esta investigación. Gracias por introducirme en un campo tan relevante y demandado en la actualidad, el cual ha despertado un profundo interés en mi.

A mi compañero David, con quien compartí las primeras fases de investigación y cuyo trabajo conjunto fue clave para plantear las bases del proyecto.

Y, por último, a mi familia, amigos y en especial a mi pareja por su constante apoyo a lo largo de todas las etapas de este trabajo. Su ánimo y comprensión me impulsaron a seguir adelante incluso en los momentos en los que el tiempo escaseaba.

Capítulo 1

Introducción

A lo largo de este capítulo, se desarrolla la introducción del Trabajo Fin de Máster. En primer lugar, se presenta la motivación del proyecto, que expone el contexto y la necesidad de la implementación de inteligencia artificial en el ámbito cultural. A continuación, se definen los objetivos del estudio, que guían el desarrollo de la propuesta. Finalmente, el documento concluye con una descripción de los contenidos y la estructura general del trabajo.

1.1. Motivación del proyecto

La motivación de este Trabajo Fin de Máster se fundamenta en la convergencia de dos fenómenos relevantes: el avance significativo de la IA generativa, en particular los modelos de lenguaje de gran tamaño (LLMs), y la creciente necesidad de los museos de innovar en sus formas de interacción con el público. La irrupción de herramientas como ChatGPT [1] ha demostrado que es posible mantener conversaciones naturales y contextuales con sistemas automatizados, lo que abre nuevas posibilidades para su aplicación en entornos culturales.

En el ámbito museístico, diversas instituciones han comenzado a explorar el potencial de la IA para enriquecer la experiencia del visitante. Por ejemplo, el Museo del Prado ha implementado herramientas de lectura aumentada basadas en IA para contextualizar sus obras y facilitar su comprensión [2]. Asimismo, el uso de tecnologías interactivas y personalizadas se ha identificado como una estrategia clave para atraer a audiencias más jóvenes y diversificar el perfil del público [3].

Este proyecto propone desarrollar una plataforma conversacional que actúe como guía virtual en el Museo Nacional Thyssen-Bornemisza [4]. El objetivo es ofrecer a los visitantes una herramienta que proporcione información contextualizada y adaptada a sus intereses, mejorando así la accesibilidad y la interacción con las obras expuestas. Al integrar la IA generativa en el entorno museístico, se busca no solo modernizar la experiencia del visitante, sino también contribuir a la reflexión sobre el papel de la tecnología en la difusión y conservación del patrimonio cultural.

1.2. Contexto del proyecto

Como se ha señalado en la sección anterior, este trabajo se origina como respuesta al creciente dinamismo en el avance de modelos de aprendizaje automático. En

la actualidad, existen una amplia variedad de modelos de IA, cada uno con objetivos específicos, que se están adaptando y desplegando en diversas instituciones y corporaciones a nivel mundial [5]. Estos modelos han experimentado un desarrollo acelerado, impulsado por avances técnicos y una mayor disponibilidad de datos, lo que ha permitido su aplicación en una gama cada vez más amplia de sectores.

Son varios los sectores específicos a los que se aplican estas nuevas técnicas, entre los que destacan el ámbito de la investigación médica, con herramientas como AlphaFold, que ha permitido transformar el análisis de proteínas mediante un sistema de IA desarrollado por Google DeepMind, entre otros.

A medida que estos modelos avanzan, la adaptación de la IA a sectores específicos se intensifica, ampliando las capacidades de automatización, análisis predictivo y toma de decisiones en tiempo real. Este fenómeno genera nuevas oportunidades de innovación, creando un panorama en constante evolución en el que las instituciones deben mantenerse competitivas, favoreciendo el auge y crecimiento de este tipo de herramientas.

1.3. Fundamentos

En esta sección se exponen los conceptos fundamentales necesarios para comprender qué son los LLMs y las aplicaciones prácticas que pueden derivarse de su uso. Con el fin de proporcionar una visión integral, se abordarán algunas de las tareas más relevantes que estos modelos son capaces de realizar. Este análisis resulta crucial para entender el rol que los modelos de lenguaje desempeñan en el desarrollo del *chatbot* propuesto, ya que establece los cimientos sobre los cuales se construye la solución presentada en este trabajo.

1.3.1. *Large Language Models (LLMs)*

Los LLMs son grandes modelos de lenguaje capaces de generar texto, entender texto o ambas opciones. Se trata de uno de los casos más prolíficos de la IA generativa en los últimos años debido a la expansión de modelos como ChatGPT, entre otros. Estos modelos permiten realizar tareas completamente diferentes, como:

- **Análisis de sentimiento:** se trata de una técnica de extracción de información basada en técnicas de procesamiento de lenguaje en la que se analiza la subjetividad de un texto dado, es decir, trata de determinar la actitud del interlocutor. Es muy útil en la obtención de retroalimentación de opiniones en base a comentarios.
- **Traducción de texto:** permiten traducir textos con mejor precisión que los traductores tradicionales debido a que son capaces de comprender mejor los textos.
- **Chatbots:** es uno de los usos más conocidos debido a ChatGPT de OpenAI. Permiten al usuario final interactuar a modo de conversación con el modelo de lenguaje.

Teniendo en cuenta las capacidades anteriores, con este tipo de modelos como base podemos generar las siguientes herramientas:

- **Asistentes virtuales:** son sistemas más avanzados que permiten mantener conversaciones naturales con el usuario, ofreciendo respuestas personalizadas y, en muchos casos, integrándose con servicios o dispositivos para realizar tareas

(como programar recordatorios, enviar correos o consultar información). A diferencia de un *chatbot* de uso, un asistente virtual no tiene por qué ser en forma de chat, y suele usarse para un conjunto de tareas concretas.

- **Respuesta a la base de conocimientos:** la técnica, que a menudo se denomina procesamiento del lenguaje natural intensivo en conocimiento, se refiere a los LLMs que pueden responder a preguntas específicas a partir de la información en los archivos digitales.
- **Clasificación de textos:** mediante la agrupación en clústeres, los LLMs pueden clasificar textos con significados o sentimientos similares. Los usos incluyen medir la opinión de los clientes, determinar la relación entre los textos y buscar documentos.
- **Generación de código:** los LLM dominan la generación de código a partir de indicaciones en lenguaje natural. Uno de los ejemplos más prolíficos es Github Copilot.
- **Generación de texto:** al igual que la generación de código, la generación de texto puede completar oraciones incompletas o escribir la documentación del producto.

1.3.2. Chatbot

Un *chatbot* es un programa informático que permite la interacción persona - ordenador a través de la simulación de una conversación mediante texto o voz. Para el desarrollo de este tipo de herramientas se utilizan principalmente técnicas de procesamiento de lenguaje natural (NLP).

En los últimos años, el desarrollo de modelos de lenguaje avanzados ha permitido una evolución significativa en la forma en que se diseñan e implementan estos sistemas. Un claro ejemplo es ChatGPT -ya citado con anterioridad- un modelo desarrollado por OpenAI que se basa en la familia de modelos GPT (*Generative Pre-trained Transformer*). Esta arquitectura, entrenada sobre grandes volúmenes de texto y ajustada mediante técnicas de aprendizaje supervisado y por refuerzo, ha demostrado una gran capacidad para mantener conversaciones naturales, comprender preguntas complejas y generar respuestas contextualizadas.

A diferencia de los *chatbots* tradicionales, que solían estar basados en reglas fijas o en árboles de decisión, los modelos como ChatGPT, Llama, Gemini, entre otros, permiten una interacción mucho más flexible, donde el comportamiento del sistema no está completamente predefinido, sino que se adapta a cada conversación en función del contexto y la intención del usuario. Esta nueva forma de generar *chatbots* ha reducido de forma significativa los esfuerzos de diseño manual, ha ampliado los dominios de aplicación y ha mejorado notablemente la experiencia de usuario.

1.4. Objetivos

El objetivo de este Trabajo Fin de Máster es investigar y demostrar el potencial de los LLMs como base para construir sistemas conversacionales aplicados al ámbito cultural. A través de este trabajo se busca explorar cómo la IA generativa puede utilizarse para diseñar nuevas formas de interacción entre los visitantes y los contenidos de un museo.

Uno de los fines principales es validar la capacidad de los LLMs para adaptarse a contextos específicos mediante técnicas actuales de ajuste, integración de conocimiento y diseño de conversación. El trabajo pretende analizar en qué medida estas herramientas permiten desarrollar chatbots capaces de responder de forma coherente, relevante y útil a las preguntas de los usuarios en un entorno tan concreto como el de una institución cultural.

El contexto elegido, el del Museo Nacional Thyssen-Bornemisza, se toma como caso representativo para aplicar estas tecnologías. La intención es mostrar cómo un modelo conversacional puede integrarse dentro de una experiencia de visita, ofreciendo al usuario información accesible, personalizada y contextualizada sobre la colección del museo y otros aspectos relacionados con su recorrido.

En última instancia, el trabajo tiene como propósito contribuir a la reflexión sobre el papel de la IA en espacios culturales. Se plantea cómo este tipo de soluciones pueden transformar la forma en que los visitantes se relacionan con el contenido, mejorando la accesibilidad y fomentando una interacción más activa y dinámica durante la visita.

1.5. Contenidos del documento

El presente documento se estructura en los siguientes capítulos, en los que se abordan los distintos aspectos del proyecto:

- **Introducción (Capítulo 1):** Se describe la motivación del proyecto, el contexto en el que se desarrolla el mismo, los aspectos funcionales relacionados con LLMs y chatbots, así como los objetivos y motivaciones del proyecto.
- **Estado del arte y trabajos relacionados (Capítulo 2):** En este capítulo, se detallan los trabajos previos y el estado actual en áreas relevantes como LLMs y las herramientas asociadas a la realización de este trabajo.
- **Desarrollo del proyecto (Capítulo 3):** Este capítulo está compuesta por los siguientes apartados:
 - **Obtención de los datos:** explicación de los métodos de obtención de los datos utilizados para la realización del proyecto, además de la presentación de los mismos.
 - **Diseño de la aplicación:** explicación detallada del diseño conceptual y funcional de la aplicación.
 - **Arquitectura de datos:** descripción de la arquitectura empleada para gestionar y procesar los datos.
 - **Ingesta de datos:** detalle de los procesos y técnicas utilizados para la ingesta de datos.
 - **Implementación:** implementación técnica de la solución.
- **Demostración (Capítulo 4):** En este capítulo se presenta una demostración tanto visual como de los resultados obtenidos de varias conversaciones realizadas con el *chatbot*.
- **Evaluación (Capítulo 5):** Se presenta una evaluación detallada del rendimiento del *chatbot*, incluyendo los resultados de las pruebas y métricas utilizadas para medir su eficacia y precisión en las tareas propuestas.

- **Impacto y resultados (Capítulo 6):** En este capítulo, se revisan los impactos del proyecto, tanto desde el punto de vista técnico como práctico. También se discuten los resultados obtenidos, sus implicaciones y posibles mejoras futuras.
- **Apéndices**
Finalmente, se incluyen varios apéndices con información adicional de interés y otros detalles relevantes para la comprensión completa del proyecto.

Capítulo 2

Estado del arte y herramientas

En este capítulo se presentará el estado del arte en el contexto de los LLMs, algunos trabajos relacionados y las herramientas necesarias para desarrollar un proyecto de estas características.

2.1. Fundamentos de los LLMs

Un LLM es un modelo de aprendizaje profundo basado en transformadores [6]. Los transformadores son una clase de redes neuronales [7] que cuentan con la capacidad de autoatención. El concepto de atención se refiere a un mecanismo que permite a los modelos de NLP ponderar de manera selectiva diferentes partes de una secuencia de entrada, lo que mejora tanto la comprensión como la generación del lenguaje. Este enfoque permite que los modelos se concentren en la información relevante y establezcan relaciones contextuales complejas a lo largo de toda la secuencia de texto, sin depender del orden secuencial en el procesamiento de los datos.

Este cambio en la arquitectura permite manejar secuencias largas de texto y resolver problemas de pérdida de contexto y eficiencia al procesar simultáneamente toda la secuencia de palabras. Los transformadores logran esto mediante lo que se denomina atención de cabezas múltiples (*multi-head attention*), un mecanismo que permite al modelo gestionar distintas partes de la entrada de manera simultánea, mejorando así la comprensión del contexto y las relaciones entre las palabras. Este enfoque, además de mejorar la precisión, aumenta significativamente la eficiencia computacional, lo que facilita el entrenamiento de modelos más grandes y complejos.

2.1.1. Enfoques

Los LLMs adoptan diversos enfoques para abordar tareas específicas. Mientras algunos se basan en técnicas de preentrenamiento y ajuste fino (*fine-tuning*), otros enfatizan la capacidad de razonamiento intrínseca de los modelos.

2.1.1.1. Preentrenamiento

Es la forma más fundamental de entrenamiento y es equivalente al entrenamiento de otros modelos de aprendizaje automático. Consiste en entrenar un modelo desde parámetros inicializados aleatoriamente, ajustándolos para predecir el siguiente *token* en una secuencia dada y realizado mediante aprendizaje autosupervisado. Este método de aprendizaje consiste en generar etiquetas implícitas a partir de datos no

estructurados mediante el uso de grandes corpus de texto que permiten al modelo aprender la estructura y sintaxis del lenguaje, así como el conocimiento contenido en los textos.

2.1.1.2. Ajuste fino

El ajuste fino [8] implica la adaptación de un modelo preentrenado a tareas específicas. Este proceso se centra en modificar los pesos de las últimas capas del modelo, preservando las representaciones generales adquiridas durante el preentrenamiento y especializándolas para tareas particulares. Esta técnica mejora la eficiencia computacional y reduce el tiempo de entrenamiento, ya que no requiere un entrenamiento completo desde cero.

Existen varias técnicas de ajuste fino, entre las que destacan:

- *Low Rank Adaptation (LoRA)*: esta técnica introduce matrices de bajo rango entrenables en las capas del modelo original, lo que permite adaptar el modelo sin modificar directamente sus pesos principales. LoRA reduce significativamente la cantidad de parámetros necesarios para el ajuste fino, haciendo posible trabajar con modelos grandes incluso con recursos limitados.
- *Quantized Low Rank Adaptation (QLoRA)*: es una extensión de LoRA que utiliza cuantización de los pesos (por ejemplo, a 4 bits) para reducir aún más el uso de memoria. QLoRA permite ajustar modelos de gran tamaño con una huella computacional muy baja.
- *Prefix Tuning*: en lugar de ajustar los pesos del modelo, esta técnica añade una secuencia fija de vectores (los “prefijos”) al inicio de cada capa de atención. Estos vectores se entrenan específicamente para la tarea deseada, permitiendo un ajuste fino eficiente y modular.
- *P-Tuning*: similar al prefix tuning, P-tuning aprende representaciones continuas que se insertan en la entrada del modelo. Utiliza embeddings continuos en lugar de tokens discretos para optimizar el rendimiento en tareas específicas, especialmente útil en modelos de lenguaje.
- *Prompt Tuning*: en lugar de modificar el modelo o insertar nuevos vectores en las capas internas, esta técnica entrena un conjunto de prompts (entradas de texto o vectores embebidos) que guían al modelo hacia la tarea deseada. Es una de las técnicas más ligeras y se centra únicamente en modificar las entradas al modelo.

2.1.1.3. Reinforcement learning from human feedback (RLHF)

El entrenamiento mediante aprendizaje por refuerzo [9] es una técnica de optimización aplicada a modelos de aprendizaje automático, en la que se emplean estrategias de retroalimentación humana para mejorar las respuestas del modelo. Este enfoque se basa en la elaboración de un ranking de las respuestas generadas por el modelo, el cual es evaluado por expertos humanos. Mediante este proceso, el modelo aprende a identificar las respuestas correctas e incorrectas, optimizando así su desempeño. Aunque esta técnica es efectiva para mejorar la precisión y relevancia de las respuestas, su principal limitación reside en la necesidad de intervención humana, lo cual puede ser un factor que incremente la carga de trabajo y los costos asociados.

2.1.1.4. *Adapters*

La técnica de *adapters* [10] es un enfoque similar al *fine-tuning*, pero con una diferencia significativa en su implementación. En lugar de modificar todos los parámetros del modelo preentrenado, los *adapters* consisten en la adición de nuevas capas a la red neuronal, que serán las únicas que se ajusten durante el proceso de entrenamiento posterior. Este enfoque permite conservar la estructura original del modelo preentrenado, aprovechando su conocimiento general, al tiempo que se introduce especialización adicional a través de las capas añadidas. A diferencia del ajuste fino tradicional, que se limita a la modificación de las últimas capas del modelo, los *adapters* pueden insertarse en diversas capas de la red neuronal, ofreciendo una mayor flexibilidad y eficiencia en la adaptación del modelo a tareas específicas.

2.1.1.5. *Prompt engineering*

La ingeniería de *prompts* constituye una de las técnicas más básicas y fundamentales en la interacción con modelos de lenguaje. Consiste en la formulación de secuencias de texto de manera que el modelo pueda interpretar y ejecutar las instrucciones proporcionadas. Los *prompts* son fragmentos de texto en lenguaje natural que describen las acciones o comportamientos que se esperan del modelo, permitiendo dirigir su desempeño sin necesidad de modificar su estructura interna. Esta técnica resulta esencial en el manejo eficiente de los modelos de lenguaje, ya que permite maximizar la capacidad de respuesta del modelo a través de un control explícito y claro de las entradas, sin requerir un reentrenamiento exhaustivo del modelo.

Esta técnica puede ser aplicada añadiendo indicaciones o ejemplos. Para el segundo caso se establecen los siguientes términos:

- **Zero Shot:** no se añade ningún ejemplo de entrada al modelo.
- **One Shot:** se añade un ejemplo de entrada-salida al modelo.
- **Few Shot:** se añade más de un ejemplo de entrada-salida al modelo.

2.1.1.6. *Retrieval augmented generation (RAG)*

Esta técnica se basa en proporcionar al modelo contexto relevante relacionado con la pregunta planteada, con el fin de generar una respuesta adecuada y coherente.

El principio subyacente de esta técnica radica en el uso de modelos de *embeddings* [11], un tipo de modelos de lenguaje diseñados para identificar relaciones semánticas entre palabras y convertir consultas en vectores. Al transformar las palabras de la consulta en vectores, es posible trabajar en un espacio vectorial, lo que permite medir distancias entre los vectores correspondientes.

Una vez transformada la consulta en un vector, se procede a comparar este vector con fragmentos de texto o *chunks* almacenados en una base de datos vectorial. Se selecciona el fragmento cuyo vector tenga la menor distancia semántica con respecto al vector de la consulta. Los *chunks* que presenten la menor distancia semántica, es decir, aquellos con mayor congruencia semántica, se incorporan al *prompt* del modelo de lenguaje, proporcionando así contexto adicional. Gracias a la capacidad de razonamiento del modelo de lenguaje, este puede generar una respuesta a la consulta original, fundamentada en el contexto suministrado.

Esta técnica resulta particularmente útil en escenarios en los que es necesario consultar información dinámica, que está sujeta a cambios frecuentes, o cuando se ma-

nejan datos privados o limitados. También es eficaz en situaciones en las que no se dispone de la capacidad computacional o la cantidad suficiente de datos para entrenar un modelo completo. No obstante, esta técnica es sensible a diversos factores, como el tamaño de los *chunks*, el rendimiento del modelo de *embedding* y la naturaleza del texto introducido. Por lo tanto, en escenarios complejos, la simple aplicación de la distancia semántica puede resultar insuficiente, lo que requiere la implementación de técnicas adicionales para optimizar este proceso.

2.1.1.7. Sistemas multiagentes

Este tipo de sistemas se fundamentan en el concepto de agentes [12]. Un agente puede definirse como un sistema que utiliza la capacidad de razonamiento de los modelos de lenguaje para tomar decisiones y emplear diversas herramientas de acuerdo con las necesidades específicas planteadas por una consulta inicial. La integración de múltiples herramientas y agentes da lugar a los sistemas multiagente, los cuales permiten la creación de agentes especializados en tareas particulares. Estos agentes, al interactuar entre sí, son capaces de determinar cuál de ellos debe ser utilizado en función de la naturaleza y requisitos de la respuesta solicitada.

2.1.2. Modelos actuales

En esta sección, se analizarán algunos de los modelos más destacados en la actualidad, destacando sus características técnicas, su rendimiento y sus principales aplicaciones. Se presentarán tanto modelos LLMs como modelos de *embeddings*, ya que ambos han sido necesarios para el desarrollo del proyecto.

2.1.2.1. LLMs

- **GPT-4.5 (OpenAI)**: Presentado el 27 de febrero de 2025, GPT-4.5 es el modelo de lenguaje más grande y avanzado de OpenAI hasta la fecha. Entrenado principalmente mediante aprendizaje no supervisado, complementado con *fine tuning* supervisado y RLHF, GPT-4.5 mejora significativamente en la comprensión del contexto, la alineación con la intención del usuario y la inteligencia emocional. Aunque OpenAI no ha revelado oficialmente el número exacto de parámetros, estimaciones independientes sugieren que GPT-4.5 cuenta con aproximadamente 12,8 billones de parámetros totales, con alrededor de 600 mil millones de parámetros activos, utilizando una arquitectura *mixture-of-experts* (MoE) similar a la de GPT-4. Esta arquitectura permite que solo una fracción de los parámetros se active durante la inferencia, optimizando el uso de recursos computacionales [13]. Destaca por ofrecer conversaciones más naturales y empáticas, reduciendo las alucinaciones en comparación con versiones anteriores. Aunque no está diseñado específicamente para tareas de razonamiento lógico o matemático, muestra mejoras notables en benchmarks como MMMLU (85.1 %) y MMMU (74.4 %) [14, 15, 16].
- **o3 (OpenAI)**: Lanzado en abril de 2025, este modelo representa el avance más significativo de OpenAI en razonamiento multimodal. Incorpora una "cadena de pensamiento privada" que le permite planificar y razonar antes de generar respuestas, mejorando la precisión en tareas complejas. Destaca en *benchmarks* como SWE-Bench Verified [17] (71.7 % frente al 48.9 % de su predecesor o1) y CodeElo [18] (Elo 2727 frente a 1891). Además, puede integrar imágenes en su proceso de razonamiento, permitiendo analizar y manipular visualmente ele-

mentos como esquemas o notas manuscritas. Este modelo también accede de manera autónoma a herramientas como navegación web, ejecución de código Python, análisis de archivos y generación de imágenes, optimizando la resolución de problemas complejos [19]. Se desconoce el número de parámetros ya que la empresa OpenAI no ha publicado información al respecto.

- **LLaMA 4 (Meta):** La nueva serie de modelos de Meta, presentada en abril de 2025, presenta tres modelos: Llama 4 Behemoth, compuesto por 288B de parámetros activos y 16 expertos formando un total de 2T parámetros; Llama 4 Maverick, formado por 17B de parámetros activos, 128 expertos formando un total de 400B de parámetros y Llama 4 Scout, compuesto por 17B de parámetros activos y 16 expertos formando un total de 109B de parámetros. Esta nueva serie introduce la arquitectura MoE, ya utilizada por los modelos GPT-4.5. Cada uno de los tres modelos propuestos se especializa en tareas concretas: Behemoth enfocado en la creación de modelos destilados, Maverick se trata de un modelo multimodal y Scout presenta ventanas de contexto ampliadas de 10M de tokens. Como las anteriores series del modelo, destacada por ser de código abierto y por su eficiencia en tareas de NLP [20].
- **DeepSeek-R1 (DeepSeek):** Lanzado por la empresa china DeepSeek en enero de 2025, este modelo ha captado atención por su eficiencia y capacidad de razonamiento en tareas complejas, como matemáticas y programación. Ofrece resultados competitivos a un costo significativamente menor, siendo de código abierto y accesible para la comunidad científica. Los modelos DeepSeek-R1 disponen de opciones con 1.5B, 7B, 8B, 14B, 32B y 70B parámetros y se trata de un modelo destilado entrenado utilizando técnicas de *reinforcement learning* [21].
- **Grok 3 (X):** Grok 3 es un modelo multimodal lanzado en febrero de 2025 por la empresa X, capaz de analizar imágenes, comprender lenguaje natural y responder preguntas complejas. Su integración con la plataforma X (anteriormente conocida como Twitter) permite una interacción directa con contenidos en redes sociales, ofreciendo respuestas contextualizadas basadas en información actualizada. Grok 3 se caracteriza por su enfoque en la personalización de la experiencia del usuario, adaptándose a diferentes estilos de conversación y necesidades específicas [22]. Se desconoce el número de parámetros ya que la empresa X no ha publicado información al respecto.
- **Claude (Anthropic):** los modelos de la serie Claude, introducidos por el laboratorio Anthropic, presentan varias versiones entre las que destaca Claude 3.7 Sonnet. Este modelo, presentado en febrero de 2025, introduce un enfoque único para el razonamiento, integrando la capacidad de razonamiento profundo y respuestas rápidas dentro del mismo modelo. En lugar de separarlos como en otros modelos del mercado, Claude 3.7 Sonnet permite a los usuarios alternar entre un modo estándar, similar a Claude 3.5 Sonnet, y un "modo de pensamiento extendido", donde el modelo reflexiona antes de responder, mejorando su desempeño en tareas complejas como matemáticas, física, seguimiento de instrucciones y codificación [23]. Se desconoce el número de parámetros ya que la empresa Anthropic no ha publicado información al respecto.

2.1.2.2. Modelos de *embeddings*

- **Text-embedding-3-large (OpenAI):** El modelo *text-embedding-3-large* de OpenAI es una versión avanzada de su serie de modelos de incrustación de texto,

diseñada para generar representaciones vectoriales de alta calidad con hasta 3072 dimensiones. Este modelo ha demostrado un rendimiento superior en tareas de evaluación estándar, como MIRACL y MTEB [24, 25], superando a sus predecesores en precisión y capacidad de captura semántica [26].

- **Text-embedding-ada-002 (OpenAI):** Este modelo de OpenAI es una solución versátil y rentable para generar incrustaciones de texto. Con una dimensión de salida de 1536 y una longitud máxima de entrada de 8191 tokens, *text-embedding-ada-002* unifica capacidades previamente distribuidas en cinco modelos distintos, abarcando tareas como búsqueda de texto, similitud semántica y búsqueda de código. Su diseño eficiente ofrece un rendimiento comparable al modelo *text-similarity-davinci-001*, pero a un costo significativamente menor, lo que lo convierte en una opción popular para aplicaciones que requieren incrustaciones de texto de alta calidad [26].
- **Nomic-embed-text (Nomic AI):** El modelo *nomic-embed-text*, desarrollado por Nomic AI, es un codificador de texto con una longitud de contexto extendida que ha superado a modelos como *text-embedding-ada-002* y *text-embedding-3-small* de OpenAI en tareas tanto de contexto corto como largo. Este modelo destaca por su capacidad para manejar documentos extensos, proporcionando representaciones vectoriales precisas y consistentes, lo que lo hace especialmente útil en aplicaciones que requieren comprensión de textos largos y complejos [27].
- **Multilingual-e5-large (Intfloat):** Este modelo, desarrollado por intfloat y basado en la arquitectura XLM-RoBERTa, cuenta con 24 capas y genera vectores de 1024 dimensiones. Está diseñado para tareas como recuperación de textos, similitud semántica y minería de texto bilingüe en más de 100 idiomas. Entrenado mediante aprendizaje contrastivo débilmente supervisado, ha demostrado un rendimiento destacado en *benchmarks* como Mr. TyDi y MTEB [28, 25]. Su capacidad para procesar múltiples idiomas lo hace especialmente útil en aplicaciones de búsqueda semántica y clasificación multilingüe [29].

2.2. Trabajos relacionados

Son varias las publicaciones y trabajos relacionados, tanto aplicando enfoques de IA generativa, como aplicados a mejorar la experiencia en entornos museísticos. En esta sección se presentarán algunos de los trabajos relacionados aplicados a museos, así como algunos trabajos enfocados en arquitecturas basadas en IA generativa.

2.2.1. Trabajos relacionados en entornos museísticos

2.2.1.1. Modelos de lenguaje como sistema de recomendación

Georgios Trichopoulos, Markos Konstantakis, Georgios Alexandridis y George Caridakis exploran el uso de LLMs aplicados a entornos museísticos utilizados como sistema de recomendación [30]. En el estudio presentado, se utiliza un enfoque de *fine-tuning* sobre modelos de lenguaje para incorporar información contextual e instrucciones personalizadas que permiten a los modelos ofrecer recomendaciones adaptadas a las preferencias del usuario. Estos sistemas son capaces de tener en cuenta factores como el tiempo disponible para la visita, la localización dentro del museo y otros parámetros contextuales, con el fin de proporcionar al visitante una experiencia más personalizada y enriquecedora.

2.2.1.2. *Chatbots como herramientas de interacción en museos*

Otro ejemplo destacado de aplicaciones de IA generativa en entornos museísticos es el del *Centre Pompidou*, que junto a *Ask Mona* ha implementado un *chatbot* basado en IA generativa para ofrecer recomendaciones personalizadas y responder a preguntas frecuentes de los visitantes [31]. Este *chatbot*, disponible a través de plataformas como *Centre Pompidou Chatbot* [32] y *Ask Mona* [33], se ha diseñado para interactuar en varios idiomas y proporcionar información tanto textual como en formato de audio, lo que enriquece la experiencia del visitante. *Ask Mona*, en particular, es un asistente que provee contenido detallado de más de 130 obras de la colección permanente del museo, junto con recursos adicionales sobre otras piezas no cubiertas por el modelo de lenguaje. Según la descripción proporcionada por *Ask Mona*, el sistema busca atraer a una audiencia más joven, que a menudo prefiere no ser acompañada por un guía tradicional, y desea explorar el museo de manera autónoma, con contenido curado por expertos en arte.

El *chatbot* del *Centre Pompidou* permite a los usuarios interactuar con el modelo de manera intuitiva. Los visitantes pueden tomar fotografías de las obras de arte, y el *chatbot*, mediante tecnologías de reconocimiento de imágenes, identifica la obra y ofrece información contextual, como anécdotas históricas, detalles sobre el autor y otros elementos exclusivos. Además de su función informativa, este asistente virtual también responde a inquietudes prácticas del visitante, tales como horarios, tarifas, opciones para niños, y la localización de objetos perdidos, lo que hace que la visita sea más cómoda y eficiente.

2.2.1.3. *Digital Curator Training Tool Box (DCbox)*

El proyecto DCbox [34] se enfoca en la digitalización del patrimonio cultural, destacando la importancia de la colaboración entre diversas disciplinas y el desarrollo de habilidades especializadas para afrontar los desafíos actuales del sector. En este proyecto, se ha enfatizado la necesidad de incorporar tecnologías de manera eficiente en las instituciones culturales, lo que incluye la creación de exposiciones híbridas y el uso de aplicaciones que combinan lo físico con lo digital, conocidas como *phygital*. Este enfoque subraya la relevancia de contar con profesionales que puedan adaptarse a un entorno digital en constante cambio, promoviendo tanto el acceso como la conservación de los bienes culturales mediante nuevas estrategias tecnológicas.

Este proyecto es clave para comprender la evolución de la IA en los museos, ya que pone de manifiesto la creciente demanda de competencias profesionales en el sector cultural y su conexión con la adopción de tecnologías emergentes, con el fin de mejorar la experiencia de los visitantes y asegurar la sostenibilidad de las instituciones.

2.2.1.4. *VirtuWander*

VirtuWander [35] presenta un sistema basado en LLMs diseñado para mejorar las interacciones multimodales en tours virtuales, especialmente en museos. El sistema está diseñado para ofrecer guías personalizadas, adaptándose a las necesidades de los usuarios mediante un marco de diseño basado en tres categorías: el momento de la guía, la información ambiental necesaria y las tareas que el usuario desea realizar. Este enfoque permite que *VirtuWander* facilite diversas formas de retroalimentación multimodal en museos virtuales, integrando conocimientos específicos del dominio y contexto.

El sistema fue evaluado a través de un estudio de usuarios en un museo virtual simulado, demostrando que mejora la experiencia del tour al ofrecer comunicación personalizada y asistencia informada.

2.2.1.5. GPTs aplicados a entornos museísticos

OpenAI presenta la posibilidad de crear GPTs customizados por la comunidad. Esto ha abierto la puerta a la creación de modelos aplicados a distintos enfoques, entre ellos, a entornos museísticos. Un ejemplo de ello es el caso de un GPT personalizado que se ajusta a los datos del Museo del Prado¹. Este GPT permite interactuar con obras del museo. A diferencia de la solución propuesta en este trabajo, este modelo no dispone de datos actualizados y únicamente sirve para explicar algunas de las obras disponibles en el museo. En consecuencia, este GPT se diferencia de la herramienta implementada en este proyecto en la incapacidad de responder preguntas relacionadas con actividades, recorridos o información actualizada acerca de horarios, tarifas y ubicaciones de obras.

2.2.2. Trabajos relacionados enfocados en LLMs

2.2.2.1. Mecanismos de colaboración en soluciones multiagente

Khanh-Tung Tran y colaboradores [36] realizan una revisión exhaustiva sobre los mecanismos de colaboración entre agentes basados LLMs, destacando su potencial para abordar tareas complejas de forma colectiva. Proponen un marco conceptual que clasifica estos mecanismos según actores, tipos de interacción (cooperación, competencia o una combinación de ambas), estructuras organizativas, estrategias y protocolos de coordinación.

2.2.2.2. Técnicas de RAG avanzadas

Las técnicas de RAG suponen un gran avance a la hora de aportar contexto a los LLMs, puesto que presentan una solución efectiva al incorporar información externa y actualizable; sin embargo, el RAG tradicional presenta una serie de desventajas. Yunfan Gao y colaboradores [37] exploran estrategias de optimización del proceso de indexado - estrategias de segmentación, anexos de metadatos e índice estructural -, optimización de la consulta - expansión, transformación y enrutación de la consulta - y técnicas aplicadas al modelo de *embedding* - ajuste fino y adapters -. Por otro lado, Yuntong Hu y colaboradores [38] introducen los conocidos como GRAGs (*Graph Retrieval-Augmented Generation*), un enfoque que combina información textual y topológica mediante una estrategia de divide y vencerás para recuperar subgrafos relevantes de forma eficiente. GRAG permite a los LLMs generar respuestas conscientes del contexto estructural del grafo, superando a métodos RAG convencionales en tareas que requieren razonamiento multisalto sobre grafos textuales.

2.3. Herramientas

En esta sección se abordarán las herramientas disponibles para las tareas realizadas en la realización del proyecto. Desde las herramientas para la obtención de los datos hasta aquellas necesarias para la implementación del *chatbot*. En la figura 2.1

¹<https://chatgpt.com/g/g-ZdIYrWWs4-guia-del-museo-del-prado>

se muestra de forma esquemática las herramientas que se van a presentar en esta sección.

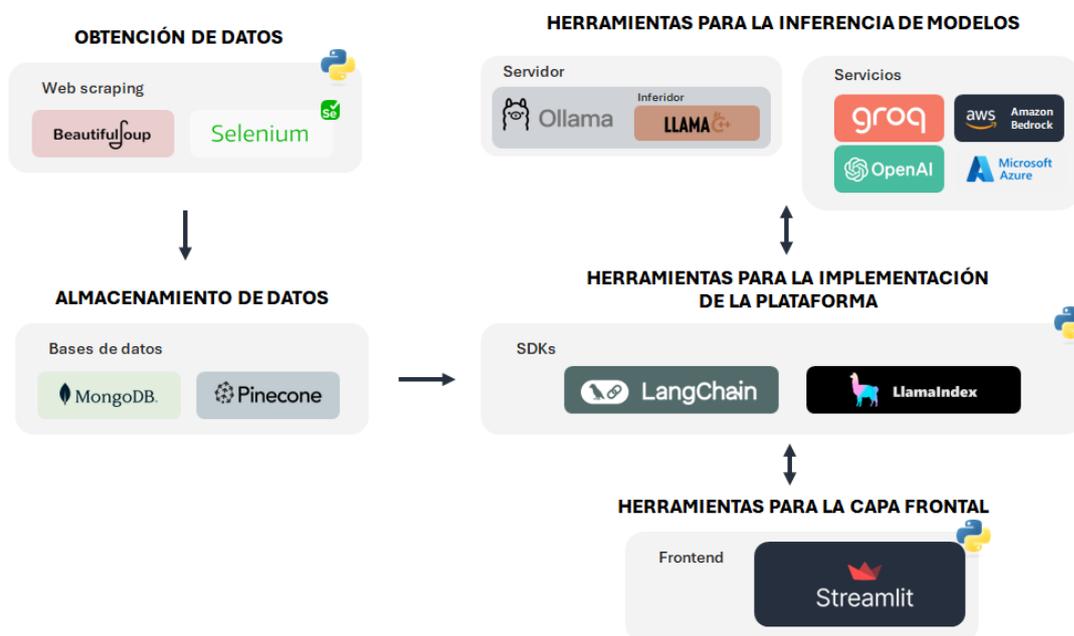


Figura 2.1: Herramientas presentadas.

2.3.1. Python

Python es un lenguaje de programación interpretado, de alto nivel y de propósito general, ampliamente utilizado en la comunidad científica y tecnológica debido a su legibilidad, versatilidad y extensa colección de bibliotecas especializadas [39]. Algunas de las bibliotecas utilizadas se abordarán en las siguientes secciones.

La decisión de utilizar Python como lenguaje para el desarrollo de la aplicación se debe principalmente a la disponibilidad de varias bibliotecas que permiten manejar LLMs como LangChain.

2.3.2. Herramientas para *web scraping*

El *web scraping* es una técnica utilizada para extraer información de sitios web de manera automática. A través de esta metodología, los datos disponibles en formato HTML o en otras estructuras web pueden ser recolectados, procesados y transformados en formatos manejables como bases de datos, hojas de cálculo o archivos CSV para su posterior análisis [40]. Esta práctica es fundamental en tareas de minería de datos, especialmente cuando la información no está fácilmente disponible a través de APIs, como es el caso.

La mayoría de herramientas disponibles para realizar esta tarea se pueden encontrar en bibliotecas de Python.

2.3.2.1. BeautifulSoup

Es una de las bibliotecas más populares para analizar y extraer datos de documentos HTML y XML. Permite navegar por la estructura del árbol DOM de las páginas para localizar y extraer el contenido deseado mediante selectores de etiquetas, clases o atributos. Esta biblioteca ha sido utilizada para la obtención de los datos disponibles en las páginas del museo estáticas, es decir, aquellas relacionadas con la información de autores y de obras [41].

2.3.2.2. Selenium

Selenium² es una herramienta poderosa para automatizar navegadores web. A diferencia de BeautifulSoup, Selenium permite interactuar con páginas dinámicas, realizar clics en botones, completar formularios y manejar sitios que utilizan JavaScript. Esta biblioteca se ha utilizado para obtener los datos provenientes de las páginas dinámicas del museo, es decir, aquellas relacionadas con recorridos y actividades.

2.3.3. Inferidor - Servidor

Un inferidor es un sistema que toma un modelo preentrenado y responde a solicitudes de entrada en tiempo real. En concreto, en lo que concierne a este proyecto, cabe destacar *Llama.cpp* como inferidor relacionado.

LLaMA.cpp es un inferidor de código abierto que realiza inferencias en varios modelos de lenguaje de código abierto, pero en especial, en aquellos de la familia de LLaMA. Este inferidor está diseñado para ejecutar estos modelos de forma eficiente en dispositivos con recursos limitados [42].

El concepto de servidor, en el contexto de LLMs, responde a una aplicación que permite acceder a modelos de lenguaje a través de solicitudes locales o remotas. En lugar de que cada cliente cargue el modelo directamente, el servidor maneja la carga del modelo y las inferencias, proporcionando una interfaz de programación de aplicaciones (API) para que otros sistemas puedan interactuar con el modelo de manera eficiente.

Ollama³ es una de estas herramientas. Ollama se trata de un servidor que permite la inferencia de modelos de lenguaje de código abierto.

El concepto inferidor-servidor se encuentra muy ligado entre sí. En este caso, las herramientas que actúan como servidor tienen implementados bajo su arquitectura inferidores. En lo que respecta al proyecto, Ollama trabaja como servidor realizando el proceso de inferencia a través de *Llama.cpp*.

2.3.4. SDKs

Un SDK (*Software development kit*)[43] en el contexto de LLMs, es un conjunto de herramientas, bibliotecas y documentación que permite a los desarrolladores integrar modelos de lenguaje en sus aplicaciones de manera sencilla. Los SDKs proporcionan una interfaz estructurada para realizar inferencias, manejar solicitudes de texto, y a menudo incluyen funcionalidades avanzadas como gestión de contexto, almacenamiento de conversaciones y manejo de datos personalizados.

²<https://www.selenium.dev/>

³<https://ollama.com/>

2.3.4.1. LlamaIndex

LlamaIndex⁴ es un *framework* diseñado para crear aplicaciones de IA generativa aumentada por contexto, basadas en LLMs, que integran agentes y flujos de trabajo [44]. Este *framework* proporciona herramientas para construir aplicaciones de LLMs con contexto aumentado, cubriendo desde la fase de prototipado hasta la producción. Entre sus funcionalidades, permite ingerir, parsear, indexar y procesar datos propios para implementar flujos de trabajo que combinan el acceso a datos privados con la generación de *prompts* para los LLMs. LlamaIndex es especialmente eficaz para aplicaciones que utilizan técnicas de RAG. Por defecto, emplea el modelo *text-embedding-ada-002* de OpenAI para generar los *embeddings*.

2.3.4.2. LangChain

LangChain⁵ es un *framework* ampliamente utilizado para crear aplicaciones basadas en LLMs. Permite desarrollar agentes, cadenas, *chatbots* y aplicaciones de RAG, entre otras muchas soluciones. Sus principales características incluyen:

- **Gestión de memoria:** a diferencia de los LLMs, que procesan cada consulta de manera independiente, LangChain permite retener información de interacciones previas, manteniendo conversaciones coherentes gracias a su capacidad para gestionar el contexto. Ofrece diferentes tipos de memoria: desde la que guarda toda la conversación, hasta versiones que retienen solo las interacciones más recientes y resumen las anteriores.
- **Chains:** una *chain* o cadena es una secuencia de operaciones conectadas en un flujo de datos, donde la salida de una operación es la entrada de la siguiente. LangChain proporciona una interfaz intuitiva y flexible para crear herramientas modulares, facilitando la construcción de *workflows* complejos.
- **Agentes:** incluye agentes preconfigurados y personalizables que simplifican su uso. Además, permite la integración con LangGraph para construir sistemas multiagente avanzados.
- **Integración con múltiples APIs y fuentes de datos:** LangChain se conecta fácilmente con una amplia variedad de APIs, lo que permite cambiar de modelos base de manera sencilla. También admite la integración con diversas fuentes de datos, como documentos y bases de datos.

LangChain se complementa con otras bibliotecas de la misma compañía, como:

- **LangGraph**⁶, para crear agentes personalizables.
- **LangSmith**⁷, una plataforma para desarrolladores que facilita la depuración, pruebas, evaluación y monitoreo de los pasos de aplicaciones implementadas en el ecosistema de LangChain.
- **LangServe**⁸, que permite desplegar APIs REST —interfaces que permiten la comunicación entre aplicaciones a través de HTTP— basadas en aplicaciones desarrolladas con LangChain.

Langchain es especialmente adecuado para:

⁴<https://www.llamaindex.ai/>

⁵<https://www.langchain.com/>

⁶<https://www.langchain.com/langgraph>

⁷<https://www.langchain.com/lagnsmith>

⁸<https://python.langchain.com/docs/langserve/>

- Aplicaciones basadas en *chatbots*, gracias a su eficaz gestión del contexto.
- *Prompts* avanzados para LLMs
- Aplicaciones que requieren acceso a múltiples fuentes de datos.

Además, cuenta con una comunidad extensa que proporciona soporte y contribuciones constantes al ecosistema.

Ambos *frameworks* presentan múltiples utilidades. Debido al enfoque del proyecto, se ha decidido optar por el uso de LangChain, ya que posee mejor conectividad con otras herramientas, mejor documentación y mayor modularidad.

2.3.5. Plataformas y servicios

En el ecosistema de los LLMs, han surgido diversas plataformas y servicios en la nube para facilitar su integración. Estos servicios permiten acceder a modelos de lenguaje sin necesidad de ser ejecutados en una máquina remota, simplemente accediendo como servicios.

2.3.5.1. Amazon Bedrock

Amazon Bedrock⁹ es un servicio de AWS que permite a los desarrolladores crear aplicaciones de IA generativa sin necesidad de gestionar infraestructuras complejas. Proporciona acceso a una variedad de modelos fundacionales de distintos proveedores, como Anthropic, AI21 Labs, Stability AI y Amazon. Además, Bedrock facilita la personalización de modelos con datos específicos y la integración con otros servicios de AWS, como S3 y Lambda, para crear soluciones escalables y seguras.

2.3.5.2. Microsoft Azure

Microsoft Azure¹⁰ es una plataforma de nube pública de Microsoft que ofrece una amplia gama de servicios para construir, implementar y gestionar aplicaciones y servicios a través de una red global de centros de datos. Azure proporciona una infraestructura escalable y flexible que permite a los desarrolladores crear, almacenar, analizar y gestionar datos y aplicaciones sin necesidad de tener sus propios servidores físicos. Además, Azure incluye recursos de IA impulsados por OpenAI, lo que permite ejecutar modelos de dicha compañía desde su servicio Azure OpenAI Service¹¹.

2.3.5.3. OpenAI API

OpenAI API¹² es un servicio que proporciona acceso a los modelos de IA de OpenAI, como GPT-4, entre otros. A través de una interfaz sencilla basada en llamadas a una API REST, permite integrar estos modelos en aplicaciones sin necesidad de entrenar modelos propios. Además, OpenAI ofrece opciones de personalización, como ajustes mediante instrucciones específicas (GPT-4 Turbo con *custom instructions*) o *fine-tuning*, entre otros.

⁹<https://aws.amazon.com/bedrock/>

¹⁰<https://azure.microsoft.com/es-es>

¹¹<https://azure.microsoft.com/es-es/products/ai-services/openai-service>

¹²<https://openai.com/index/openai-api/>

2.3.5.4. Groq API

*Groq API*¹³ es un servicio que proporciona acceso a modelos de lenguaje optimizados para alto rendimiento y baja latencia. A diferencia de otras soluciones en la nube, Groq se enfoca en ofrecer respuestas ultrarrápidas gracias a su arquitectura de aceleradores diseñados específicamente para modelos de lenguaje. Actualmente, soporta modelos de código abierto como los de Mistral, Gemma o Llama en varias de sus versiones, permitiendo integrar IA generativa en aplicaciones con costos competitivos y tiempos de respuesta reducidos. Además, ofrece servicios gratuitos diarios limitados a un número concreto de *tokens* según el modelo utilizado. En concreto para este trabajo se han utilizado la serie de modelos Llama.

2.3.6. Bases de datos

Para la implementación del sistema, es necesaria la utilización tanto de bases de datos no relacionales como vectoriales, optimizadas para la recuperación eficiente de información mediante técnicas de RAG. A continuación, se describen las principales bases de datos empleadas.

2.3.6.1. MongoDB

MongoDB es una base de datos NoSQL orientada a documentos que almacena datos en formato JSON binario (BSON), lo que permite una mayor flexibilidad en comparación con bases de datos relacionales. A diferencia de los modelos tradicionales basados en tablas y esquemas rígidos, MongoDB utiliza colecciones y documentos, lo que facilita la manipulación de datos semiestructurados y su escalabilidad horizontal [45].

2.3.6.2. Pinecone DB

Pinecone¹⁴ es una base de datos especializada en la búsqueda vectorial, diseñada para manejar grandes volúmenes de datos no estructurados y encontrar información relevante de manera eficiente mediante representaciones en forma de vectores. En lugar de almacenar datos en tablas como una base de datos relacional o en documentos como MongoDB, Pinecone indexa vectores de alta dimensión generados por modelos de aprendizaje automático, lo que permite realizar búsquedas por similitud con tiempos de respuesta muy bajos.

2.3.7. Streamlit

*Streamlit*¹⁵ es un *framework* de código abierto en Python diseñado para la creación rápida y sencilla de aplicaciones web interactivas, especialmente enfocadas en proyectos de ciencia de datos, *machine learning* y visualización de datos. Su principal ventaja es que permite desarrollar interfaces gráficas con pocas líneas de código, sin necesidad de conocimientos avanzados en desarrollo web.

Además, Streamlit permite desplegar aplicaciones fácilmente en la web sin necesidad de configurar servidores complejos. Es una herramienta ideal para la creación rápida de prototipos, *dashboards* y demostraciones interactivas en el ámbito de la IA. Para

¹³<https://groq.com/>

¹⁴<https://www.pinecone.io/>

¹⁵<https://streamlit.io/>

el contexto del proyecto, se han utilizado las herramientas que relacionadas a la creación de *chatbots*.

Capítulo 3

Desarrollo

Durante este capítulo, se abordará el proceso de desarrollo del trabajo realizado, especificando la fuente de obtención de los datos, las componentes fundamentales de Museion, la arquitectura de datos adoptada y los aspectos técnicos en cuanto a la implementación de la plataforma.

Se prestará especial atención al diseño de la arquitectura, a los modelos de lenguaje empleados y a las decisiones estratégicas tomadas a lo largo del desarrollo de Museion. Además, se describirá detalladamente el flujo de información dentro de la plataforma, abarcando desde la recuperación de datos hasta la generación de respuestas.

Por último, se discutirán los principales desafíos encontrados durante el proceso, junto con las soluciones implementadas para asegurar una experiencia conversacional eficiente y precisa.

3.1. Obtención de los datos

Como se ha señalado en el capítulo 1, este proyecto tiene como caso de uso las colecciones del Museo Nacional Thyssen-Bornemisza. Para la obtención de los datos necesarios, se ha optado por emplear técnicas de web scraping [40] aplicadas a la página web oficial del museo¹. En consecuencia, la fuente de los datos es el propio museo, lo que garantiza la fiabilidad y la autenticidad de la información utilizada en el proyecto.

El propósito de este trabajo se centra en el diseño e implementación de la arquitectura del modelo de lenguaje. Los datos disponibles en la página web del museo son de carácter dinámico, lo que implica que están sujetos a modificaciones constantes (por ejemplo, en relación con las exposiciones disponibles, las salas de exhibición, las tarifas de entrada o los recorridos sugeridos). Es por ello que los datos utilizados para este proyecto podrían estar obsoletos o desactualizados en el momento de su demostración. No obstante, el diseño de la aplicación permite la actualización continua de los datos, asegurando que, en un entorno real, la plataforma pueda mantenerse actualizada con la información más reciente.

Para la versión de demostración de este proyecto, los datos utilizados corresponden a las actualizaciones más recientes disponibles en las siguientes fechas:

¹<https://www.museothyssen.org/>

- La información acerca de obras y autores está actualizada a 18 de mayo de 2025.
- La información acerca de tarifas, exposiciones y actividades está actualizada a 18 de mayo de 2025.
- La información acerca de recorridos está actualizada a 18 de mayo de 2025.

La página web del museo está compuesta por las siguientes secciones:

- **Visita:** En este apartado se presenta información relevante sobre la visita al museo, incluyendo detalles como tarifas, planos del espacio y una descripción de las actividades y exposiciones disponibles. Además, se incluye una sección dedicada a los recorridos, donde se ofrecen itinerarios temáticos, complementados con información adicional sobre las obras que integran cada uno de estos recorridos.
- **Colección:** Esta sección se divide en cuatro subapartados. El primero, *Autores*, ofrece información sobre los artistas cuyas obras se encuentran expuestas en el museo. En el segundo apartado, *Obras Maestras*, se destacan 32 de las piezas más significativas de la colección. El tercer apartado, *Colección Permanente*, permite acceder a un catálogo completo de las obras del museo. Finalmente, el subapartado *Colección Carmen Thyssen* proporciona información sobre la cesión de parte de la colección de la familia Thyssen al museo, así como detalles sobre las obras que la conforman.
- **Exposiciones:** En esta sección se puede consultar tanto las exposiciones pasadas como las actuales.
- **Actividades:** Este apartado proporciona información sobre todas las actividades organizadas por el museo, tales como visitas guiadas, conferencias, talleres y cursos, entre otras.

Cabe destacar que los datos empleados para la demostración no corresponden a la totalidad de la información disponible en el museo, sino que constituyen una selección representativa de obras, artistas, actividades y exposiciones de relevancia significativa, para las cuales se dispone de información detallada. El conjunto completo de autores, obras, exposiciones y actividades sobre los que Museion posee información está formado por la unión de todos aquellos que cumplen alguno de los siguientes criterios:

- Los artistas que se consideren “artistas seleccionados” citados en la página web de la colección del museo², además de algunos artistas importantes de la cultura española. La lista de autores puede ser consultada en el Anexo A.1.
- Las obras que componen la sección de obras maestras de la página del museo³ -disponibles en el Anexo A.2-, todas las obras pertenecientes a los “artistas seleccionados” (véase A.1) y todas las obras que forman parte de alguno de los recorridos temáticos disponibles (véase A.3).
- Las exposiciones disponibles en la fecha de recogida de datos.
- Las actividades disponibles a fecha de la recogida de datos.

²<https://www.museothyssen.org/coleccion>

³<https://www.museothyssen.org/coleccion/obras-maestras>

- Los recorridos disponibles a fecha de la recogida de los datos - a excepción del recorrido *Pintura y dolor en las colecciones Thyssen-Bornemisza* -, mencionados en el Anexo A.3.

3.2. Diseño de la aplicación

Como se mencionó en el capítulo anterior, la página del Museo Thyssen-Bornemisza, que ha servido como modelo para este trabajo, se organiza en varias secciones claramente definidas. En consecuencia, se ha optado por estructurar la plataforma en tres subsecciones que abarcan las principales funcionalidades presentes dicho sitio web.

Cada una de estas subsecciones es accesible a través de una página de inicio que describe las funcionalidades de las herramientas correspondientes. La implementación de esta interfaz se ha llevado a cabo utilizando el *framework* Streamlit (véase 2.3.7), el cual ha facilitado el desarrollo de una interfaz acorde a la herramienta sin necesidad de diseñar manualmente cada componente. Gracias a su arquitectura declarativa y su integración con Python, ha sido posible centrarse en los aspectos técnicos del desarrollo, delegando la generación de la interfaz a este *framework*, la cual se aleja del alcance del proyecto. Esta decisión ha permitido agilizar el proceso de desarrollo y asegurar una experiencia de usuario fluida y accesible.

Así, el enfoque principal del desarrollo del trabajo ha estado en la gestión de datos, la implementación del modelo conversacional y la optimización de la recuperación de información.

En la figura 3.1 se muestra la página de inicio de la herramienta. A través de esta interfaz, es posible acceder a las distintas secciones disponibles, las cuales se detallarán a lo largo de esta sección.

3.2.1. Sección “Información del museo”

La sección “Información del museo” constituye una de las tres funcionalidades principales de la herramienta desarrollada. Se trata de un asistente conversacional, implementado en forma de *chatbot*, diseñado para responder a las preguntas más frecuentes que los visitantes puedan tener antes de su experiencia en el museo.

Mediante esta herramienta, los usuarios pueden obtener información sobre aspectos esenciales de su visita, tales como los precios de entrada, las actividades disponibles, las exposiciones temporales y permanentes, los horarios de apertura y cierre, la ubicación de las obras dentro del museo y los descuentos aplicables. Gracias a los modelos de lenguaje empleados, el sistema es capaz de interpretar las preguntas formuladas en lenguaje natural y encontrar las fuentes de datos necesarias para responder de forma precisa y personalizada a las preguntas de los visitantes.

Esta funcionalidad proporciona un acceso rápido y eficiente a la información provista en la página web sin necesidad de navegar por diversas secciones del sitio web. Al estar integrada en Museion, permite a los usuarios interactuar de manera intuitiva y obtener respuestas inmediatas, mejorando así su experiencia antes y durante la visita al museo.

La decisión de crear una herramienta específica para responder a estas consultas se basa en varias razones técnicas y operativas. En primer lugar, esta implementación permite una posible integración en la página web del museo, ofreciendo a los usuarios

Museion

Museo Nacional Thyssen-Bornemisza

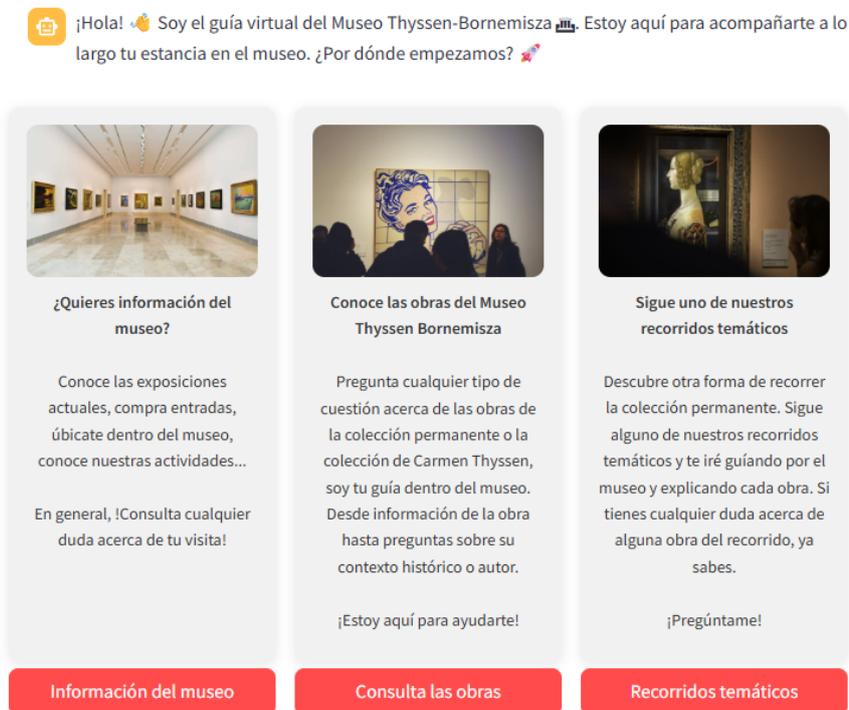


Figura 3.1: Página de inicio de Museion.

una forma dinámica y accesible de obtener la información necesaria para su visita con una simple pregunta, en lugar de navegar por todas las distintas pestañas. De este modo, el *chatbot* actúa como un asistente interactivo al que el usuario puede consultar sus dudas sobre la futura visita, mejorando su experiencia de navegación.

Además, la creación de una herramienta independiente optimiza el funcionamiento global del sistema. Al aislar esta funcionalidad específica, se reduce la complejidad del sistema de categorización de las preguntas del usuario (un aspecto que se abordará en las secciones siguientes), lo que facilita la gestión de las consultas. Esta simplificación también implica una reducción en el número de *tokens* utilizados, lo cual no solo mejora la latencia de la herramienta, sino que también reduce los costos operativos asociados al procesamiento de cada consulta. Esto permite mantener un rendimiento eficiente, tanto para esta funcionalidad como para las demás funciones de Museion.

3.2.2. Sección “Consultar obras”

La sección “Consultar obras” ha sido diseñada para proporcionar a los usuarios información detallada, contextualizada y personalizada sobre las piezas de la colección del museo. Esta funcionalidad permite responder a consultas relacionadas con las descripciones de las obras, sus autores, el contexto histórico en el que fueron creadas, detalles específicos, así como su localización dentro del museo. Gracias a su inte-

gración con el *chatbot*, los visitantes pueden acceder en tiempo real a información precisa y adaptada a sus intereses, lo que enriquece significativamente su experiencia durante la visita.

Al igual que en la sección anterior, existen razones fundamentadas para desarrollar esta herramienta como una funcionalidad independiente. La principal reside en su interés como herramienta aislada que sirva como guía virtual interactiva durante el recorrido por el museo. Dado que está orientada a proporcionar información detallada sobre las obras en tiempo real, permite a los usuarios ampliar su conocimiento de forma inmediata mientras exploran la colección. Esta característica transforma la herramienta en un asistente activo que acompaña al visitante, brindando respuestas personalizadas sobre cada obra o autor en función de las consultas formuladas. A diferencia de las guías tradicionales, esta solución se adapta dinámicamente a las necesidades informativas del usuario, sintetizando la información y contextualizándola según el interés particular del momento.

Asimismo, y al igual que en la sección anterior, la creación de una herramienta específica para esta función contribuye a la optimización del sistema en su conjunto. La especialización de esta funcionalidad permite reducir la complejidad de los procesos de interpretación y categorización de preguntas, al tiempo que mejora la eficiencia del modelo conversacional. Como resultado, se minimizan las latencias y se reducen los costos computacionales asociados, lo cual repercute positivamente tanto en el rendimiento de esta sección como en el de las demás funcionalidades de la plataforma.

3.2.3. Sección “Recorridos temáticos”

La sección “Recorridos temáticos” representa la tercera y última funcionalidad de Museion, y tiene como objetivo ofrecer a los usuarios una visita guiada interactiva a través de los distintos itinerarios temáticos disponibles en el museo. Esta herramienta ha sido diseñada para facilitar una exploración estructurada y contextualizada de la colección, permitiendo a los visitantes aproximarse a las obras desde enfoques diversos.

El asistente comienza presentando al usuario las distintas opciones de recorridos temáticos, brindándole la posibilidad de seleccionar aquel que resulte de mayor interés. Una vez realizada la elección, el sistema proporciona una introducción a la temática seleccionada, destacando su relevancia dentro del conjunto museístico. A continuación, el visitante es guiado virtualmente sala por sala, descubriendo las obras incluidas en el recorrido, cada una de ellas contextualizada en función de la narrativa correspondiente.

Con el fin de enriquecer la experiencia del usuario, el asistente también permite formular preguntas específicas sobre cada obra incluida en el recorrido. De este modo, si el visitante desea profundizar en algún aspecto particular o resolver dudas concretas, puede obtener respuestas inmediatas y precisas. Esta funcionalidad transforma la herramienta en un auténtico guía virtual, mejorando significativamente la accesibilidad a la información y ofreciendo una forma innovadora, autónoma y personalizada de recorrer el museo.

3.3. Arquitectura de datos

En esta sección se abordarán las decisiones e implementaciones relacionadas con la arquitectura de datos utilizada en Museion.

Dado que es fundamental mantener actualizada la información sobre obras, tarifas, horarios, actividades, exposiciones y recorridos, resulta imprescindible diseñar una arquitectura de datos eficiente, escalable y adaptable en el tiempo. La tipología de los datos manejados exige un acceso rápido y óptimo, la capacidad de expandirse en el tiempo y la posibilidad de actualización continua para garantizar que los usuarios siempre dispongan de información veraz y vigente.

Para responder a estos requerimientos, se ha optado por una solución basada en dos tipos de bases de datos complementarias: una base de datos no relacional, encargada de gestionar información estructurada y semiestructurada de manera flexible, y una base de datos vectorial, utilizada para optimizar la recuperación de información a partir de modelos de lenguaje. Esta combinación permite equilibrar rendimiento, escalabilidad y precisión en la gestión de datos dentro de la plataforma.

3.3.1. Base de datos no relacional

Para la gestión eficiente de la información estructurada y semiestructurada de Museion, se ha optado por el uso de una base de datos no relacional, específicamente MongoDB (véase 2.3.6.1). Esta base de datos almacena toda la información relacionada con obras, autores, recorridos, tarifas y exposiciones.

El uso de MongoDB responde a la necesidad de contar con una arquitectura de datos escalable y adaptable que permita almacenar información en un formato dinámico y fácilmente actualizable. Esta base de datos también cumple una función fundamental dentro del sistema, ya que actúa como respaldo para la base de datos vectorial. Mientras que la base de datos vectorial se especializa en la recuperación de información mediante búsquedas semánticas avanzadas, MongoDB es utilizada para realizar consultas directas cuando no es necesario un procesamiento semántico, como en la obtención de información concreta sobre precios, horarios, exposiciones y actividades. Además, también almacena la información acerca de recorridos.

Gracias a esta separación de responsabilidades, se optimiza el rendimiento del sistema, asegurando que las consultas estructuradas sean rápidas y eficientes, al tiempo que se mantiene un respaldo fiable para la recuperación de datos y actualizaciones de los mismos.

Para esta base de datos, se definen las siguientes colecciones que la conforman.

3.3.1.1. Colección de artistas

La colección de artistas contiene información acerca de todos los artistas disponibles en el museo a fecha de la extracción de datos comentada en la Sección 3.1. Los campos de los que dispone esta colección pueden ser consultados en el Cuadro 3.1.

Campo	Descripción
<code>artist_url_ref</code>	URL de referencia del artista en la página web del museo
<code>artist_name</code>	Nombre del artista
<code>artist_description</code>	Descripción del artista disponible en la página del museo
<code>artist_born_date</code>	Lugar de nacimiento
<code>artist_born_place</code>	Año de nacimiento
<code>artist_death_date</code>	Lugar de fallecimiento
<code>artist_death_place</code>	Fecha de fallecimiento
<code>artist_id</code>	Identificador del artista

Cuadro 3.1: Campos de la colección `thyssen_artists`

El identificador especificado en el campo `artist_id` permite identificar al artista de forma única. Se trata de un código de tres dígitos hexadecimales que permite almacenar un total de 4096 artistas. Este código identificador es utilizado para definir la fuente de origen de los datos de la base de datos vectorial que se explicará en la siguiente sección.

3.3.1.2. Colección de obras

La colección de obras contiene información de todas las obras -tanto expuestas como no expuestas y pertenecientes a la colección permanente o a la colección de Carmen Thyssen- presentadas en la página web del museo a fecha de la extracción de datos comentada en la Sección 3.1. Los campos de los que dispone esta colección pueden ser consultados en el Cuadro 3.2.

Campo	Descripción
<code>artwork_url_ref</code>	URL de referencia de la obra en la página web del museo
<code>artwork_name_title</code>	Título de la obra
<code>artwork_artist_name</code>	Nombre del autor de la obra
<code>artwork_location</code>	Sala de exposición de la obra (o <i>Not exhibited</i>)
<code>artwork_artist_ref</code>	URL de referencia del autor en la página web del museo
<code>artwork_date</code>	Fecha de realización de la obra
<code>artwork_canvas_size</code>	Tamaño de la obra
<code>artwork_canvas_type</code>	Tipo de técnica utilizada
<code>artwork_description</code>	Descripción de la obra disponible en la página del museo
<code>artwork_id</code>	Identificador de la obra

Cuadro 3.2: Campos de la colección `thyssen_artworks`

El identificador especificado en el campo `artwork_id` funciona igual que el identificador de artista pero con un código hexadecimal de cuatro cifras, lo que permite identificar un total de 65536 obras.

3.3.1.3. Colección recorridos

La colección de recorridos contiene todos los recorridos presentes en la página web del museo a fecha de la extracción de datos comentada en la Sección 3.1. Los campos de los que dispone esta colección pueden ser consultados en el Cuadro 3.3.

Campo	Descripción
<code>tour_url</code>	URL del recorrido en la página web del museo
<code>tour_title</code>	Nombre del recorrido
<code>tour_size</code>	Número de obras que componen el recorrido
<code>short_tour_description</code>	Descripción corta del recorrido
<code>long_tour_description</code>	Descripción extensa del recorrido
<code>tour_artworks</code>	<i>Array</i> de obras del recorrido
<code>tour_image</code>	URL de imagen portada del recorrido
<code>tour_summary</code>	Resumen del recorrido
<code>tour_id</code>	Identificador del recorrido

Cuadro 3.3: Campos de la colección `thyssen_tours`

Por cada recorrido existe un *array* de obras que lo componen, almacenado en el campo `tour_artworks`. Cada elemento del *array* es una obra perteneciente al recorrido y con la información asociada al contexto del recorrido al que pertenece. Los datos almacenados para cada obra se describen en el Cuadro 3.4

Campo	Descripción
<code>artwork_author</code>	Autor de la obra
<code>artwork_name</code>	Nombre de la obra
<code>artwork_location</code>	Sala en la que se encuentra la obra
<code>artwork_description</code>	Descripción de la obra en el contexto del recorrido

Cuadro 3.4: Campos de las obras contenidas en `tour_artworks`

El identificador especificado en el campo `tour_id` funciona igual que los dos identificadores anteriores, pero con un código hexadecimal de dos cifras, lo que permite identificar un total de 256 recorridos.

3.3.1.4. Colección de tarifas

La colección de tarifas almacena la información acerca de tarifas de entradas del museo a fecha de extracción de los datos citada en la Sección 3.1. Los campos disponibles son los descritos en el Cuadro 3.5.

Campo	Descripción
<code>fee_name</code>	Tipo de tarifa
<code>price</code>	Cuantía a pagar por la tarifa
<code>conditions</code>	Condiciones para ser aplicada la tarifa

Cuadro 3.5: Campos de la colección `thyssen_fees`

3.3.1.5. Colección de actividades

La colección de actividades almacena la información acerca de las exposiciones y actividades del museo a fecha de extracción de los datos comentada en la Sección 3.1. Los campos disponibles son los descritos en el Cuadro 3.6.

Campo	Descripción
<code>activity_name</code>	Nombre de la exposición o actividad
<code>description</code>	Descripción de la exposición o actividad
<code>details</code>	Detalles de la exposición o actividad
<code>price</code>	Cuantía a pagar por la exposición o actividad
<code>buy_url</code>	URL de compra de la entrada para la exposición o actividad

Cuadro 3.6: Campos de la colección `thyssen_activities`

3.3.2. Base de datos vectorial

Para la gestión de la información que requiere búsquedas semánticas avanzadas, es decir, aquellos procesos en los que se utilizan técnicas de RAG (véase 2.1.1.6), se ha implementado una base de datos vectorial. Esta tecnología permite almacenar y recuperar información en formato de vectores de *embeddings*, facilitando así la integración de los datos con el LLM. La base de datos utilizada ha sido PineconeDB (véase 2.3.6.2), ya que permite crear índices y hacer búsquedas de forma gratuita, además de utilizar técnicas de *reranking* de forma automática [46], lo que mejora notablemente la búsqueda semántica.

El uso de una base de datos vectorial es esencial para el correcto funcionamiento de Museion, ya que permite almacenar fragmentos de texto en forma de vectores numéricos. Estos fragmentos incluyen descripciones detalladas de obras de arte, biografías de autores y otra información relevante, los cuales pueden ser utilizados como contexto en las consultas realizadas al LLM. Para generar estos vectores, se han empleado modelos de *embeddings*, que convierten el texto en representaciones vectoriales que extrapolan su significado semántico a un espacio vectorial.

Esta base de datos vectorial proporciona contexto al LLM para responder preguntas relacionadas con autores, obras y movimientos artísticos, además de permitir búsquedas semánticas, facilitando que el usuario encuentre la localización de algunas obras seleccionadas a partir de descripciones visuales de las mismas.

Dentro de la base de datos vectorial podemos encontrar dos índices principales. Ambos índices tienen una dimensión de vector de 3072 y utilizan la similitud coseno como medida de distancia. La similitud coseno, definida en la fórmula 3.1, permite medir la similitud entre dos vectores tomando los valores entre $[-1,1]$, -1 si los vectores son contrarios entre sí y 1 si son vectores idénticos.

$$\frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|} \quad (3.1)$$

Las razones de utilizar estas medidas se deben a que el modelo de *embedding* utilizado ha sido `text-embedding-3-large` de OpenAI y este modelo utiliza estas dimensiones y métricas.

Por otro lado, se han realizado pruebas usando los siguientes modelos de *embeddings*, los cuales no han concluido en resultados satisfactorios por los siguientes motivos:

- **nomic-embedding (v1.5 Ollama)**: los resultados obtenidos utilizando este modelo presentan notables problemas a la hora de realizar *embeddings* en español. En el artículo en el que se detalla la implementación del modelo [47], se especifica que ha sido entrenado únicamente con datos en inglés, por lo que se asume y se comprueba posteriormente que este modelo no funciona correctamente para textos en español.
- **multilingual-e5-large**: se han realizado pruebas con el modelo, pero al disponer de una secuencia de *tokens* demasiado corta, se ha limitado notablemente el tamaño de los pedazos de texto almacenados. Al necesitar ingresar textos demasiado largos, se ha descartado este modelo para la implementación final.

3.3.2.1. Índice de descripciones

El índice de descripciones contiene la información de obras que posteriormente servirá de contexto al LLM para responder al usuario. Los datos que sirven para embeber este índice provienen de los datos almacenados en la base de datos de MongoDB. Las descripciones provistas son las siguientes:

- **Descripción de obras de la colección `thyssen_artworks`**: la descripción de las obras, almacenada en el campo `artwork_description`.
- **Descripción de obras de la colección `thyssen_tours`**: la descripción de las obras pertenecientes a un recorrido - las obras del *array* `tour_artworks` -, almacenada en el campo `artwork_description`.
- **Descripción de artistas de la colección `thyssen_artists`**: la descripción de los artistas, almacenada en el campo `artist_description`.

Estas tres tipos de descripciones se dividen en trozos de texto por cada párrafo del texto original, o en caso de que un párrafo sea demasiado extenso, en un máximo de 800 *tokens* con un solapamiento de una ventana de 200 *tokens*.

Las decisiones adoptadas en relación con el método de segmentación (*chunking*) se fundamentan en la observación de que los párrafos, en su totalidad, contienen información relevante que podría perderse o distorsionarse al fragmentarlos en unidades más pequeñas. Esta fragmentación puede dar lugar a omisiones de contenido significativo o incluso a la generación de alucinaciones en las respuestas producidas por el modelo. Asimismo, textos excesivamente extensos pueden introducir ruido en los vectores generados por el modelo de *embedding*, lo que incrementa la probabilidad de obtener resultados inexactos en las búsquedas semánticas.

Finalmente, cabe señalar que en la construcción del índice se han incorporado metadatos, los cuales se recogen en el Cuadro 3.7.

Campo	Descripción
<code>artist_name</code>	Nombre del autor de la obra
<code>artwork_date</code>	Fecha de realización de la obra
<code>artwork_id</code>	Identificador del texto del que procede la información
<code>artwork_name</code>	Nombre de la obra
<code>exposed_room</code>	Sala en la que se encuentra expuesta la obra
<code>is_exposed</code>	Booleano que indica si la obra se encuentra expuesta
<code>permanent_collection</code>	Booleano que indica si la obra pertenece a la colección permanente

Cuadro 3.7: Metadatos del índice de descripciones

La inclusión de metadatos resulta fundamental para la implementación de técnicas avanzadas de RAG, las cuales serán detalladas en la sección siguiente. Estos metadatos permiten aplicar filtros sobre los fragmentos de texto, facilitando así la selección contextual más pertinente durante el proceso de recuperación. En particular, el identificador del texto (`artwork_id`) se ha diseñado como una clave compuesta que integra los identificadores de la obra, el artista y el recorrido (tour) al que pertenece la descripción, siguiendo el formato especificado en la siguiente expresión:

```
artwork_id+artist_id+tour_id
```

Existe un caso especial en el que el identificador de tour no está especificado, que corresponde con la descripción general (aquella no asociada a ningún tour y disponible en la página web del museo). Para este caso se toma el identificador de tour 00.

Por ejemplo, para la descripción asociada a los siguientes parámetros:

- **Obra:** *Habitación de hotel* (0392)
- **Autor:** *Edward Hopper* (11A)
- **Tour:** *Un día en Nueva York* (04)

el identificador del texto sería 039211A04

Finalmente, con el objetivo de identificar de forma unívoca cada fragmento de texto asociado a una obra dentro de la base de datos vectorial, se ha asignado a cada *chunk* un identificador propio. Este identificador se construye a partir del identificador del texto original, seguido de un sufijo `:n`, donde `n` indica la posición del *chunk* en el orden secuencial del texto. Por ejemplo, en el caso de la descripción presentada anteriormente, el identificador `039211A04:2` hace referencia al tercer fragmento (o *chunk*) de la descripción de la obra *Habitación de hotel*, del artista *Edward Hopper*, asociada al tour *Un día en Nueva York*.

3.3.2.2. Índice de localizaciones

El índice de localizaciones contiene descripciones sobre los aspectos visuales de un conjunto de obras seleccionadas (véase A.2). En este caso, los datos que componen el índice no proceden de la base de datos no relacional, sino que han sido generados

mediante el modelo ChatGPT a partir de imágenes de las obras, utilizando un *prompt* específico.

Cada fragmento de texto (*chunk*) incorporado en este índice contiene dos campos de metadatos: *artwork-name* y *artwork-artist*. Estos metadatos cumplen dos funciones fundamentales. En primer lugar, permiten realizar consultas sobre la base de datos MongoDB, que alberga la información dinámica, con el fin de recuperar la localización asociada a cada obra. En segundo lugar, estos metadatos proporcionan contexto al modelo de lenguaje (LLM) para informar al usuario del nombre y autor de la obra consultada. Esto resulta especialmente útil en aquellos casos en los que la entrada del usuario presenta ambigüedades, errores ortográficos o imprecisiones en los nombres de obras o artistas. En tales circunstancias, el LLM es capaz de realizar una comparación semántica entre los nombres introducidos por el usuario y los registros existentes, obteniendo resultados que no serían posibles mediante métodos tradicionales de comparación de cadenas.

3.4. Ingesta de datos

Para embeber la base de datos vectorial con los datos que se han especificado en la sección anterior, es necesario un proceso de ingesta de datos.

Mediante este proceso se introducen los datos necesarios a la base de datos vectorial. Para ello se ha optado por las siguientes decisiones de implementación. Cabe destacar que como se ha comentado en la sección anterior, para la ingesta de datos se ha tomado un tamaño de *chunk* de 800 *tokens* con un solapamiento de una ventana de 200 *tokens* y se ha utilizado el modelo `text-embedding-3-large` de OpenAI con las dimensiones vectoriales citadas anteriormente.

3.4.1. Ingesta en índice de descripciones

Para la ingesta de datos de descripciones, es decir, el índice que proporcionará contexto al LLM para generar respuestas pertinentes acerca de obras y autores, se han considerado las descripciones de los autores referenciados en el Anexo A.1, las cuales están almacenadas en una base de datos MongoDB. Asimismo, se incluyen las obras que conforman el conjunto de obras maestras, descritas en el Anexo A.2, así como aquellas que forman parte de los recorridos temáticos detallados en el Anexo A.3.

En el caso de las obras incluidas en los recorridos temáticos, se ha obtenido también una descripción general de cada una, independiente del contexto específico del recorrido en el que se insertan.

Cada una de estas descripciones ha sido segmentada en fragmentos de texto conforme a las especificaciones previamente mencionadas. Posteriormente, los fragmentos han sido transformados en vectores mediante el modelo de *embedding* seleccionado. Finalmente, estos vectores han sido ingresados en el índice correspondiente para su procesamiento y recuperación eficiente.

Cabe mencionar que los metadatos asociados a cada obra han sido obtenidos durante la obtención de datos de MongoDB.

3.4.2. Ingesta en índice de localizaciones

Para la ingesta de datos en el índice de localizaciones, se ha optado por un enfoque basado en la interacción con el modelo ChatGPT. En este caso, se ha utilizado

el siguiente *prompt*: “Si fueras un visitante de un museo y no supieras nada acerca de esta obra, pero quisieras buscarla, ¿cómo la describirías? Aquí tienes una descripción:” acompañado de la descripción de la obra obtenida de la página web del museo.

El texto generado por el modelo ha sido cuidadosamente revisado, al que se le ha añadido la información relevante, como el nombre de la obra, el autor correspondiente, así como el movimiento artístico o las características estilísticas asociadas a la misma. Posteriormente, las descripciones revisadas han sido convertidas en vectores mediante un modelo de *embedding* y, finalmente, se han ingresado en el índice correspondiente para su posterior consulta y recuperación.

3.5. Implementación de Museion

En esta sección se analizará la implementación de Museion. Aquí se detallan todas las integraciones relacionadas con los LLMs y la aplicación de diversas herramientas al caso de uso específico.

A lo largo de la sección, el análisis se realizará siguiendo la estructura de Museion, desglosando cada uno de sus componentes. Cada parte presenta una implementación distinta, adaptada a su función específica dentro del sistema. En algunos casos, estas implementaciones pueden solaparse o tener implementaciones similares, aunque con matices que las especializan en su tarea particular.

Por último, cabe destacar que son varios los LLMs utilizados para cada una de las soluciones que presenta Museion. A lo largo de esta sección se especificará el caso de uso en el que se ha utilizado cada uno de los modelos.

3.5.1. Gestión de la memoria

Con el fin de garantizar que la plataforma conversacional proporcione una experiencia de usuario que emule una conversación humana, resulta esencial la incorporación de una memoria conversacional. El diseño eficiente de esta memoria es muy relevante, ya que representa una de las principales limitaciones en términos del consumo de *tokens*.

Existen múltiples enfoques para la implementación de una memoria conversacional en *chatbots*. Sin embargo, considerando que el sistema está diseñado para mantener conversaciones relativamente prolongadas, se ha optado por una arquitectura que anteponga la reducción del consumo de *tokens*.

Concretamente, se ha implementado una memoria con una capacidad máxima de 300 *tokens*. Esta memoria almacena de forma secuencial tanto los mensajes introducidos por el usuario como las respuestas generadas por el modelo. Una vez alcanzado el límite de *tokens*, se activa un proceso de resumen que permite preservar los aspectos más relevantes del diálogo manteniendo bajo control el consumo de la memoria -siempre limitada a 300 *tokens*. Para ello, se realiza una llamada a un modelo de lenguaje —en este caso, *Llama 3.1 8B*— al que se le suministra el resumen acumulado hasta el momento junto con los nuevos mensajes. El modelo se encarga entonces de generar un nuevo resumen que integre la información reciente, permitiendo que el sistema avance en la conversación sin pérdida total de contexto.

Este enfoque garantiza que se mantenga una representación coherente de la conversación completa sin superar en ningún momento el umbral de *tokens* establecido.

No obstante, se reconoce como principal limitación la posible pérdida de detalles específicos durante los procesos de resumen, especialmente en interacciones largas y complejas. A pesar de ello, esta solución se considera la más viable dadas las restricciones impuestas por la ventana de contexto de los modelos utilizados y la naturaleza de las interacciones previstas en el proyecto.

Cabe destacar que esta estrategia de gestión de memoria ha sido aplicada en todos los *chatbots* y agentes descritos a lo largo de este capítulo.

3.5.2. Chatbot de información del museo

El *chatbot* de atención al visitante ha sido diseñado a partir de un enfoque *prompt-based*, en el que se incorpora toda la información necesaria para responder a consultas relacionadas con el funcionamiento del museo. Este enfoque consiste en proporcionar al modelo de lenguaje un *prompt* predefinido que integra los datos relevantes, permitiéndole generar respuestas precisas y contextualizadas ante preguntas sobre tarifas, horarios, actividades o exposiciones temporales.

Dicha información es obtenida dinámicamente a través de un controlador de MongoDB, que actúa de puente entre el modelo y la base de datos no relacional del museo. Esta base contiene información actualizada sobre los elementos anteriormente mencionados, lo que garantiza que las respuestas generadas por el asistente reflejan siempre el estado más reciente de la oferta del museo. La arquitectura utilizada se muestra en la Figura 3.2.

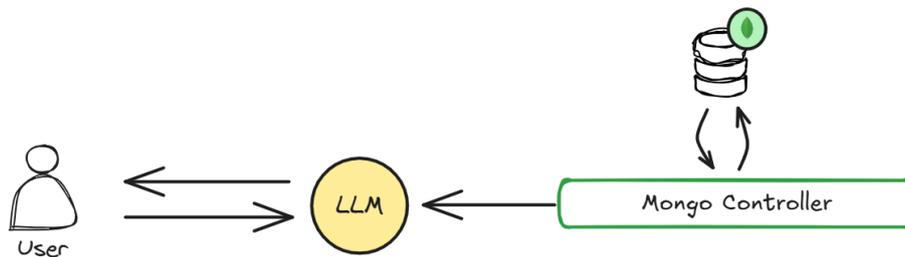


Figura 3.2: Diagrama de flujo del *chatbot* de información del museo.

La decisión de optar por esta arquitectura, en lugar de implementar un sistema basado en agentes —como se describe en secciones posteriores— responde a varios motivos. En primer lugar, el volumen de información requerido para responder a este tipo de consultas es relativamente reducido, por lo que no resulta necesario activar mecanismos complejos de coordinación entre múltiples agentes. En segundo lugar, las preguntas dirigidas a este *chatbot* tienden a ser de carácter repetitivo o ambiguo, con referencias cruzadas a distintos bloques de información (por ejemplo, horarios y exposiciones en una misma consulta), lo que hace más eficiente el uso de un *prompt* global en lugar de una delegación entre agentes especializados.

Por último, el coste computacional en términos de *tokens* de entrada no representa una desventaja significativa frente a la alternativa multiagente. Aunque esta última introduce un aumento en el número total de *tokens* debido a la coordinación entre agentes, dicho incremento resulta comparable al volumen de *tokens* requeridos por el *prompt* utilizado en esta solución monolítica.

3.5.3. Chatbot de información de obras

El asistente conversacional orientado a la consulta de obras artísticas ha sido desarrollado mediante una arquitectura basada en un sistema multiagente. En el núcleo de esta arquitectura se encuentra un agente principal, cuya función es gestionar el flujo conversacional, interpretar la intención del usuario y delegar las consultas entre un conjunto de agentes especializados.

Cada uno de estos agentes secundarios está diseñado para cumplir las siguientes funcionalidades específicas dentro del sistema:

- **Router:** nodo encargado de redirigir el flujo conversacional hacia los distintos agentes especializados según la naturaleza de la consulta del usuario.
- **Description:** agente especializado en responder preguntas relacionadas con obras, autores, así como consultas complementarias sobre contextos históricos, movimientos artísticos, entre otros aspectos vinculados al ámbito expositivo.
- **Location:** agente encargado de asistir al visitante en la localización de obras dentro del museo. Permite la búsqueda a partir de nombres de autores, títulos de obras o movimientos artísticos, incluso cuando estos contienen errores ortográficos. Asimismo, admite la identificación de obras a partir de descripciones proporcionadas por el usuario.
- **Other:** agente orientado a dirigir al usuario hacia otros recursos disponibles en el entorno del museo, como su página web, espacios físicos específicos o herramientas complementarias dentro de Museion, en los casos en que la consulta excede el alcance del sistema multiagente.
- **Non related):** agente responsable de gestionar aquellas consultas que se consideran fuera del ámbito temático del museo o que contienen contenidos restringidos.

A continuación, en la Figura 3.6, se presenta un diagrama de flujo que ilustra la arquitectura del sistema.

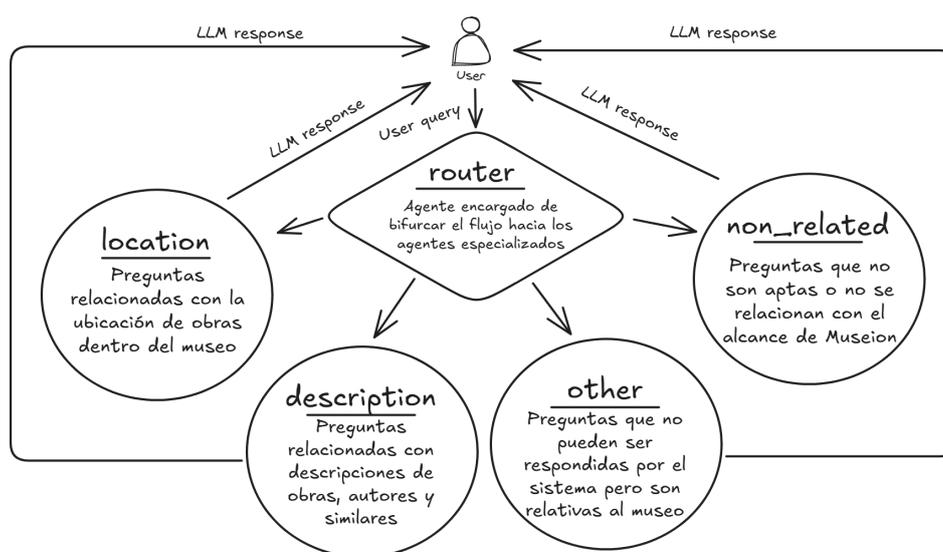


Figura 3.3: Diagrama de flujo del sistema multiagente.

Cada uno de estos agentes será descrito en detalle en las secciones siguientes.

Para permitir la comunicación y el intercambio de información entre los distintos nodos, se ha implementado un mecanismo denominado estado del agente. Este estado consiste en una estructura común que contiene una serie de campos accesibles tanto por el agente principal como por los agentes especializados que conforman el sistema multiagente. Los campos almacenados en dicha estructura son los siguientes:

- **Mensaje:** *string* que contiene la respuesta final del agente.
- **Query:** *string* que contiene la *query* del usuario.
- **Query type:** *string* con el tipo de consulta, que posteriormente servirá para realizar la bifurcación.
- **Memoria:** objeto de tipo *ConversationSummaryBufferMemory* con la memoria del *chatbot*.
- **Información de localizaciones de obras:** lista con la información del contexto que posteriormente se le ingstará al LLM, utilizado en el agente de localización de obra.
- **LLM:** objeto de tipo *ChatGroq* que contiene el modelo de lenguaje utilizado por defecto para todo el sistema multiagente.
- **Flujo de herramientas:** lista con los pasos u agentes intermedios especificados por el agente enrutador.

3.5.3.1. Agente enrutador

Tal como se ha señalado previamente, la función principal de este agente consiste en dirigir el flujo de trabajo del sistema, distribuyéndolo entre los distintos agentes especializados en función de la naturaleza de la consulta. La Figura 3.4 ilustra el esquema de funcionamiento de este agente dentro de la arquitectura general.

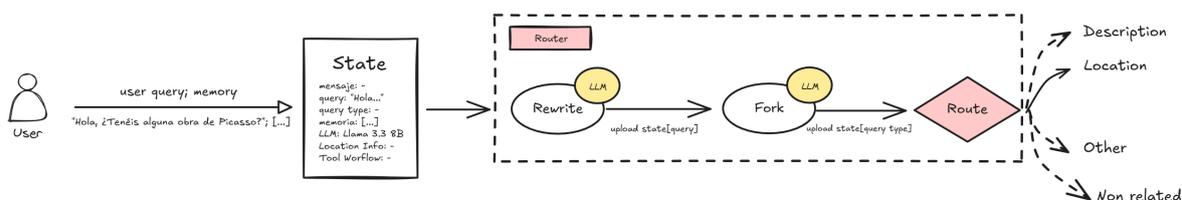


Figura 3.4: Diagrama de flujo del agente enrutador.

En esta primera etapa se introduce el *estado del agente*, descrito en la sección anterior, el cual está compuesto por una serie de campos que permiten mantener y compartir información a lo largo del flujo conversacional. Una vez inicializado dicho estado, el agente ejecuta un proceso compuesto por tres pasos principales:

1. **Reformulación de la consulta (*Rewrite*):** en esta fase inicial, la pregunta del usuario es reescrita con el objetivo de incorporar el contexto conversacional acumulado. Este proceso busca enriquecer la consulta original con información implícita o previamente mencionada, que podría haber sido omitida. Por ejemplo, si el usuario ha estado conversando sobre un artista específico y formula posteriormente la pregunta “¿Dónde están sus obras?”, el sistema carecería de los elementos necesarios (en este caso la referencia al nombre del artista) para interpretar adecuadamente la referencia. Dado que en ciertas ocasiones esta

ambigüedad puede afectar incluso a la elección del agente especializado, se ha considerado pertinente realizar esta reformulación como primer paso del flujo.

La reescritura se lleva a cabo mediante una llamada al LLM predeterminado del agente (`llama-3.3-70b-versatile`, una versión del modelo Llama 3.3 de 70B de parámetros ofrecido por la plataforma Groq), utilizando como contexto la memoria almacenada en el estado del agente. Una vez obtenida la nueva formulación, se actualiza el campo `query` del estado.

El modelo predeterminado del agente, mencionado anteriormente, hace referencia al modelo de lenguaje por defecto empleado en los casos en los que se requiere una inferencia al LLM y no se ha observado un rendimiento superior al utilizar modelos de menor tamaño. A lo largo de esta memoria, el concepto de modelo predeterminado se utilizará para designar esta versión de Llama.

2. **Clasificación de la consulta (*Fork*):** a partir de la nueva `query`, se realiza una segunda llamada al LLM (en este caso, al modelo `llama-3.1-8b-instant`, una versión del modelo Llama 3.1 de 8B de parámetros ofrecida por la plataforma Groq) cuya función es determinar el tipo de consulta planteada por el usuario. El resultado de esta inferencia se almacena en el campo `query_type` del estado del agente.
3. **Derivación al agente especializado (*Route*):** finalmente, en función del valor almacenado en `query_type`, el flujo se bifurca hacia el nodo correspondiente, es decir, hacia el agente especializado encargado de dar respuesta a la consulta planteada por el visitante.

3.5.3.2. Agente localizador

Este agente especializado tiene como función principal proporcionar información sobre la localización de obras a los visitantes. Permite realizar búsquedas basadas en nombres de obras o autores, incluso cuando contienen errores ortográficos o gramaticales. Asimismo, admite consultas por temáticas o movimientos artísticos, así como búsquedas basadas en descripciones visuales de las obras.

Es importante señalar que este agente dispone exclusivamente de información relativa a las denominadas *obras maestras* del museo, tal y como se especifica en la página web oficial de la institución.

Una vez que el flujo es dirigido hacia este agente —es decir, cuando se determina que la consulta del usuario se refiere a la localización de una obra dentro del museo—, la `query` es transformada en un vector utilizando el modelo de *embedding* empleado por la base de datos vectorial, descrito en la sección 3.3.2.

Posteriormente, se calcula la distancia semántica entre dicho vector y los *chunks* del índice de localizaciones. A través de esta comparación, se pueden identificar similitudes semánticas entre la consulta del usuario y diversas características presentes en las descripciones almacenadas:

- **Descripciones visuales:** si el usuario realiza preguntas relacionadas con atributos visuales de las obras, el sistema puede identificar correspondencias semánticas entre dichas consultas y los textos descriptivos. Por ejemplo, una consulta como: “¿Dónde se encuentra la obra en la que aparece una mujer con sombrero vestida de negro?” podría coincidir semánticamente con textos como: “... *en el*

cuadro aparece una mujer vestida de negro que dispone de un sombrero de copa y un bastón ...” o “... *la mujer vestida de negro posa delante de un paisaje ...*”.

- **Movimientos artísticos o cualidades de la obra:** además de los aspectos visuales, las descripciones también incluyen información sobre movimientos artísticos o características formales de las obras, como el estilo pictórico. Así, una consulta como: “¿Hay expuesta alguna obra impresionista pintada en puntillismo?” puede coincidir semánticamente con descripciones como: “... *esta obra, perteneciente a la etapa impresionista del autor y pintada siguiendo la técnica de puntillismo ...*” o “... *esta obra pertenece al neoimpresionismo ...*”.
- **Título de la obra o nombre del artista:** finalmente, los textos almacenados también contienen el nombre del autor y el título de la obra, lo cual permite recuperar resultados relevantes incluso cuando los nombres están escritos con errores ortográficos. De este modo, el sistema es capaz de identificar obras y artistas con nombres similares o cercanos al introducido por el usuario.

Una vez identificadas las obras potencialmente relevantes a partir de la consulta del visitante, se realiza una llamada a la base de datos MongoDB, la cual contiene la información dinámica actualizada de las obras. En esta etapa, el objetivo principal es recuperar la sala del museo en la que se encuentra expuesta cada obra.

El motivo por el cual esta información se obtiene desde MongoDB, en lugar de integrarla directamente en la base de datos vectorial, es evitar la necesidad de actualizar constantemente esta última. Dado que los identificadores de obra están presentes en los metadatos de los resultados obtenidos mediante búsqueda semántica, la recuperación de la localización desde MongoDB puede realizarse de forma inmediata y eficiente.

Finalmente, una vez obtenida la localización correspondiente, dicha información se incorpora a un *prompt* que se envía al LLM. Este *prompt* combina tanto la descripción visual recuperada desde la base de datos vectorial como la información dinámica obtenida desde MongoDB. La inclusión de la descripción visual en esta fase permite al LLM proporcionar una respuesta más rica, ya que puede añadir detalles adicionales sobre la obra y adaptar la respuesta final al tono y contenido de la consulta realizada por el visitante.

3.5.3.3. Agente descriptor

Este agente está especializado en la resolución de consultas relacionadas con obras, artistas y movimientos artísticos. Su funcionamiento se basa en la aplicación de técnicas avanzadas de RAG, las cuales se describen detalladamente a lo largo de esta sección. La fuente principal de información utilizada por este agente es el índice de descripciones almacenado en la base de datos vectorial, tal y como se menciona en la sección 3.3.2. Dicho índice está compuesto por *chunks* generados a partir de las descripciones textuales disponibles en la página web oficial del museo.

El uso de técnicas avanzadas de RAG ha sido necesario debido a dos factores principales: (i) la elevada cantidad de *chunks* disponibles en el índice, y (ii) el alto grado de solapamiento semántico entre los diferentes *chunks*, lo cual dificultaba la obtención de un contexto preciso para formular respuestas coherentes y relevantes. Entre las estrategias implementadas destacan la reescritura de consultas (*queries*) y el uso de metadatos enriquecidos.

- **Reescritura de consultas:** Con el objetivo de garantizar que la información

implícita en la pregunta original del usuario no se pierda durante el proceso de búsqueda semántica, se ha optado por reformular las consultas mediante un modelo de lenguaje intermedio (el LLM predeterminado). Esta reformulación genera tres nuevas versiones de la pregunta original, las cuales se utilizan posteriormente en la búsqueda semántica sobre el índice vectorial.

Esta estrategia responde a una limitación inherente al lenguaje natural: la tendencia a omitir información contextual relevante en formulaciones coloquiales o ambiguas. Al explicitar dicha información durante la reescritura, se mejora significativamente la precisión del vector de consulta generado y, por tanto, la relevancia de los *chunks* recuperados.

- **Uso de metadatos:** Como se ha descrito en la sección 3.3.2, el índice de descripciones contiene metadatos que se utilizan para filtrar los *chunks* en función de las características asociadas a cada obra. Este filtrado es también llevado a cabo por el modelo de lenguaje predeterminado, que transforma una pregunta no estructurada en una consulta estructurada sobre los campos disponibles en el índice.

Esta técnica permite reducir considerablemente el ruido presente en la recuperación de información. Por ejemplo, ante una consulta del tipo “¿Me podrías explicar la etapa cubista de Picasso?”, se genera una consulta estructurada que filtra los *chunks* asociados exclusivamente al artista *Picasso*. De este modo, se descartan automáticamente otros *chunks* que traten sobre el cubismo en general pero no estén relacionados con el artista mencionado, optimizando así la precisión del sistema.

A pesar de los beneficios que ofrece esta estrategia, también conlleva una serie de desafíos, tales como los posibles errores gramaticales u ortográficos a la hora de definir cualquiera de los campos del filtro o errores en la interpretación de la intención del usuario. Estas limitaciones se abordarán en detalle en el resto de esta sección.

Finalmente, utilizando las técnicas anteriormente descritas, junto con algunas que se explicarán a continuación, se obtiene la arquitectura representada en el diagrama de flujo de la Figura 3.5.

El flujo es el siguiente:

1. En primer lugar, se reescribe la pregunta original, generando un total de tres nuevas *queries* que se utilizarán para recuperar los *chunks* que servirán como contexto.
2. A continuación, se obtienen los *chunks* con mayor similitud semántica, aplicando filtros basados en metadatos para cada una de las preguntas generadas en el paso anterior. Se seleccionan un total de 10 *chunks*.
3. Como método de contingencia, en caso de que los filtros no funcionen correctamente —ya sea por errores en su aplicación o por una mala interpretación— o exista información relevante en otros fragmentos de texto que no hayan sido correctamente categorizados, se obtienen 10 *chunks* adicionales considerando únicamente la similitud semántica.
4. Con el fin de no sobrepasar la ventana de contexto, se reestructuran los *chunks* utilizados. Para ello, se eliminan aquellos fragmentos que tengan el mismo identificador de *chunk*, es decir, que hagan referencia al mismo trozo de texto. En

caso de que un mismo documento aparezca varias veces —es decir, que se hayan encontrado varios párrafos relevantes dentro de una misma fuente de información—, se considera que dicho texto es especialmente relevante y se incluyen todos los párrafos o *chunks* del mismo. Finalmente, sobre los fragmentos resultantes, se seleccionan los doce con mayor similitud semántica respecto a la pregunta original.

5. El contexto obtenido se pasa al LLM y se utiliza para generar la respuesta final.

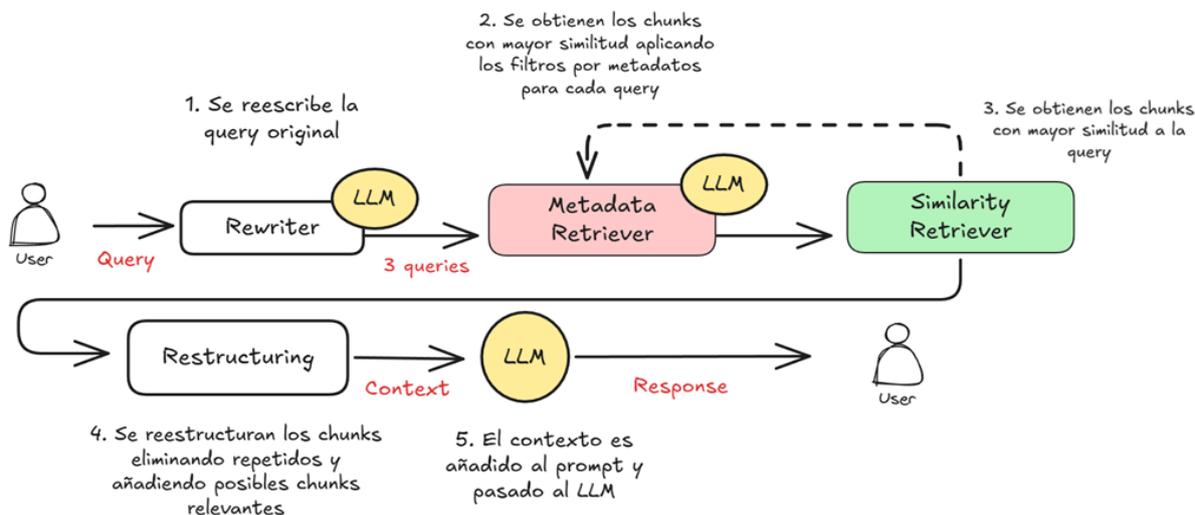


Figura 3.5: Diagrama de flujo del agente descriptor

Como se puede observar, existe la posibilidad de que el filtrado por metadatos no funcione correctamente. Como última observación, cabe mencionar la posibilidad de la existencia de errores gramaticales a la hora de definir cualquiera de los campos del filtro. Los filtros únicamente funcionan con coincidencias exactas, por lo que, en caso de producirse estos errores, no operarían correctamente.

Para mitigar este problema, en la definición de los metadatos se han especificado algunos de los valores que pueden tomar dichos campos, de modo que, en caso de que una entrada haga referencia a alguno de ellos, se pueda reescribir correctamente y el filtro funcione como es debido.

Un caso común es la escritura del autor *Pablo Picasso* como *Picasso* o *Pablo Picaso*. En estos casos, dado que se requiere una coincidencia exacta, ninguna de las dos variantes coincidiría con la búsqueda, por lo que el filtro no funcionaría correctamente.

Para evitar estos problemas, se han incorporado los autores seleccionados, descritos en la Sección 3.1, como referencia para validar y normalizar los metadatos introducidos.

3.5.3.4. Agente otras preguntas (Other)

Este agente está diseñado para responder a preguntas que se encuentran dentro del contexto del museo, pero que, por la temática a la que hacen referencia, no pueden ser respondidas directamente por Museion.

El *prompt* ha sido diseñado para que el modelo de lenguaje proporcione respuestas que remitan a los recursos disponibles del museo, como teléfonos de contacto,

la página web oficial u otras páginas de referencia relevantes. Además, se recuerda al usuario que puede consultar otras secciones de la herramienta, donde podría encontrar la información que busca.

3.5.3.5. Agente preguntas no relacionadas (*Non-related*)

Este agente está diseñado para responder exclusivamente a preguntas no relacionadas con el contexto del museo.

Principalmente, gestiona dos tipos de preguntas que quedan fuera de su ámbito:

- **Preguntas no relacionadas con el museo:** en estos casos, el agente responde de forma genérica indicando que la consulta se encuentra fuera de su alcance, y recuerda al usuario el tipo de preguntas para las que ha sido diseñado.
- **Preguntas con contenido explícito o moralmente inapropiado:** cuando el contenido de la pregunta incluye elementos explícitos o éticamente cuestionables, el agente indica que no puede responder ese tipo de consultas debido a la naturaleza del contenido.

3.5.4. Chatbot de recorridos temáticos

Como se ha comentado en secciones anteriores, la sección de *recorridos temáticos* permite realizar una visita al museo de forma contextualizada y estructurada. Para ello, se siguen una serie de recorridos diseñados por el propio museo, los cuales abordan diversas temáticas.

Para la implementación de esta parte del *chatbot*, se ha seguido la arquitectura descrita en la Figura 3.6.

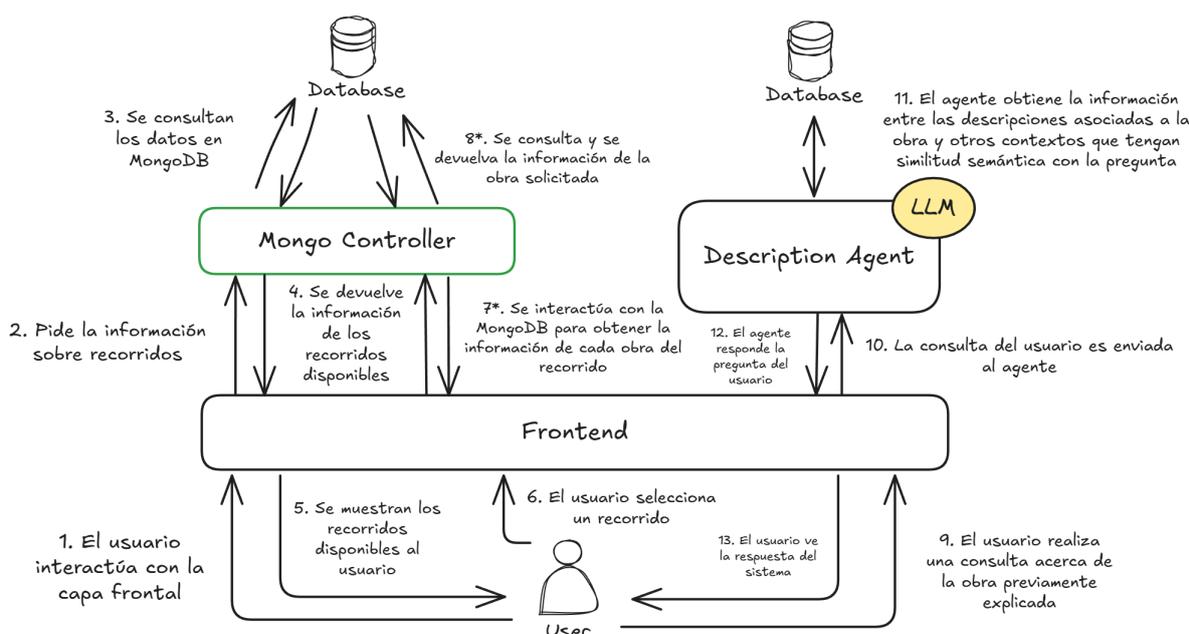


Figura 3.6: Diagrama de flujo del *chatbot* de recorridos temáticos.

3.5.4.1. Selección de recorrido temático

Una vez inicializada la sección de recorridos temáticos, se muestran los recorridos disponibles junto con una breve descripción.

Para ello, se ha implementado un controlador que interactúa directamente con la base de datos MongoDB, donde se almacena toda la información sobre los recorridos en la colección `thyssen_tours`.

A través del controlador, se accede a los títulos de los recorridos, el número de obras que los componen y su resumen. Esta información se presenta al usuario para que pueda seleccionar el recorrido de su interés.

3.5.4.2. Introducción al recorrido

Al iniciarse la sección de recorridos temáticos, el usuario interactúa con la capa frontal de Museion (paso 1 de la Figura 3.6). El *frontend* obtiene los datos haciendo una llamada al controlador (paso 2 de la Figura 3.6), y este realiza una llamada a la base de datos (paso 3 de la Figura 3.6). Finalmente, el controlador devuelve la información de los recorridos al *frontend* y el usuario visualiza los recorridos disponibles (pasos 4 y 5 de la Figura 3.6). El usuario selecciona el recorrido (paso 6 de la Figura 3.6) y tras la selección del recorrido, se realiza una consulta al controlador de MongoDB para obtener su descripción detallada (`long_tour_description`). Una vez mostrada esta descripción, se presentan una serie de mensajes predefinidos que explican al usuario el funcionamiento de la herramienta.

3.5.4.3. Descripciones de la obra

Posteriormente, el controlador de MongoDB obtiene información sobre la sala en la que se encuentra la siguiente obra del recorrido. En caso de que la obra no esté expuesta, se muestra una imagen al usuario, tal y como se ilustra en la Figura 3.7.

Cuando el usuario ha visualizado la obra, ya sea en el museo o a través de la imagen proporcionada, se le presenta la descripción correspondiente dentro del recorrido seleccionado.

Este proceso se realiza mediante los pasos 7* y 8* de la Figura 3.6. Estos dos pasos se repiten obra a obra perteneciente al tour, de ahí el asterisco mostrado en la figura.

3.5.4.4. Consultas sobre la obra

El último paso del flujo permite al usuario realizar consultas adicionales sobre la obra. Este es el único punto en el que se invoca un LLM, lo que añade una capa extra de interacción durante la visita.

La llamada al LLM se realiza a través del agente descrito en la Sección 3.5.3. La principal diferencia con respecto al *chatbot* de la sección anterior radica en la forma en que se incorpora el contexto de la obra al LLM.

Para garantizar la coherencia del diálogo, se añade al historial toda la información proporcionada al usuario durante la explicación de la obra. De esta forma, el modelo considera el contexto previo y, además, se complementa con la información adicional obtenida mediante los métodos de búsqueda semántica (RAG) descritos en la sección 3.5.3.

Desarrollo

 La siguiente obra del recorrido se titula "La Última Cena", del autor Anónimo veneciano.

 La siguiente obra del recorrido no se encuentra expuesta actualmente. Para poder disfrutarla le mostraré una imagen.



Pulse para continuar

Figura 3.7: Interfaz de usuario para obras no expuestas.

Esta fase de la conversación se ve reflejada en los pasos 9, 10, 11, 12 y 13 de la figura 3.6.

Capítulo 4

Demostración

En esta sección se hará una demostración de Museion. Para ello, se presentarán tanto la parte visual de la herramienta como algunos de los resultados de las conversaciones.

4.1. *Landing page*

La *landing page* constituye la página principal de acceso a la plataforma y cumple una función introductoria dentro de la experiencia del usuario. En ella se presentan las tres secciones principales que conforman el sistema, actuando como punto de entrada y orientación para los visitantes.

En la figura 3.1, presentada en el capítulo 3, se muestra una vista de dicha interfaz, donde puede apreciarse la distribución de los elementos. Se recuerda que los aspectos estéticos y de diseño de la interfaz se alejaban del alcance del proyecto y se han delegado a la librería *Streamlit* (véase 2.3.7).

4.2. Sección de “Información del museo”

La sección *Información del museo* presenta una interfaz conversacional basada en un chatbot de uso general. Esta herramienta busca ofrecer una experiencia de interacción natural e inmediata, facilitando el acceso a información relevante sin necesidad de navegar por múltiples páginas.

En la Figura 4.1, se muestra una captura representativa de esta interfaz, donde puede observarse la estructura del sistema conversacional y su diseño.

4.2.1. Inferencia del modelo

En esta sección se presentan las respuestas generadas por el modelo, con el objetivo de ilustrar su comportamiento ante diferentes tipos de consultas.

4.2.1.1. Pregunta 1

Pregunta: ¿Qué días se encuentra abierto el museo?

Respuesta: El Museo Nacional Thyssen-Bornemisza se encuentra abierto todos los días del año, excepto los siguientes días:

Museion

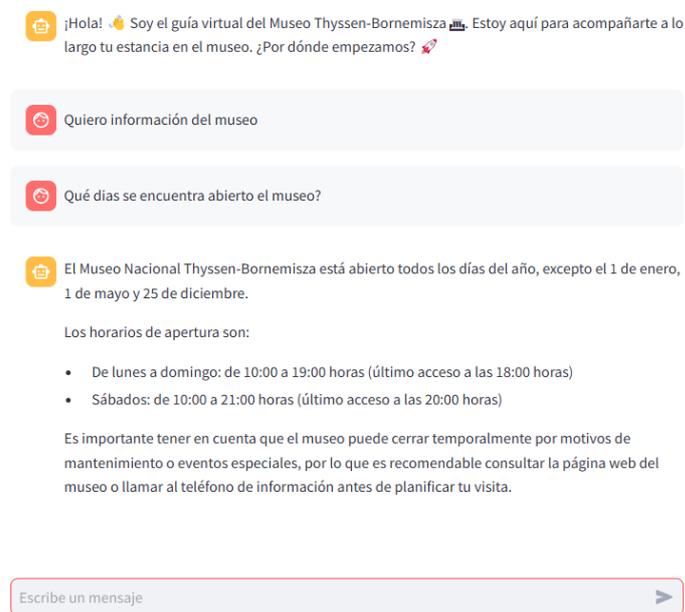


Figura 4.1: Chatbot de información del museo

- 1 de enero (Año Nuevo)
- 24 de diciembre (Nochebuena)
- 25 de diciembre (Navidad)
- Además, el museo tiene horarios especiales en algunos días del año, como los sábados, en los que se puede visitar de forma gratuita de 21:00 a 23:00 horas.

Es importante tener en cuenta que el museo puede cerrar temporalmente debido a eventos especiales o mantenimiento, por lo que es recomendable verificar la información en la página web del museo antes de planificar tu visita.

4.2.1.2. Pregunta 2

Pregunta: (Continuación de la conversación de 4.2.1.1) ¿Cuánto cuesta la entrada si tengo 23 años?

Respuesta: Si tienes 23 años, la entrada general cuesta 13€.

4.2.1.3. Pregunta 3

Pregunta: (Continuación de la conversación de 4.2.1.2) Soy estudiante, ¿existe algún descuento?

Respuesta: ¡Sí! Como estudiante, tienes derecho a un descuento. La entrada para estudiantes cuesta 9€. Recuerda que debes presentar tu carné de estudiante válido y en vigor para acceder a este descuento.

4.2.1.4. Pregunta 4

Pregunta: (Continuación de la conversación de 4.2.1.3) ¿Me puedes recomendar alguna exposición en base a mis intereses?

Respuesta: ¡Claro! Sin embargo, necesitaría saber más sobre tus intereses. ¿Te gustan los artistas clásicos, el arte moderno, el impresionismo, el arte contemporáneo...? ¿Hay algún período o estilo que te llame la atención? Algunas de las exposiciones actuales en el Museo Thyssen-Bornemisza son “Proust y las artes”, “Tarek Atoui. At-Tāriq”, “Ayako Rokkaku”, entre otras. ¿Te gustaría que te recomiende más sobre alguna de ellas?

4.2.1.5. Pregunta 5

Pregunta: (Continuación de la conversación de 4.2.1.4) Me gusta el cine y la música

Respuesta: ¡Genial! Basándome en tus gustos, te recomiendo la actividad "Sobre Vibración y Resonancia—La Colmena". Esta experiencia combina arte, música y sonidos, lo que podría ser de tu interés. Además, la actividad incluye una visita autónoma a la exposición, lo que te permitirá disfrutar del espacio y las obras de arte de manera más inmersiva.

Si prefieres algo más relacionada con el cine, podrías disfrutar de la actividad "Día Internacional de los museos 2025. Proyección de la película 'Foragers' de Jumana Manna", que es una película documental que explora la relación entre la naturaleza y la cultura.

Ambas actividades ofrecen una experiencia única y diferente en el Museo Thyssen-Bornemisza. ¿Te gustaría saber más sobre alguna de ellas?

4.2.1.6. Pregunta 6

Pregunta: (Continuación de la conversación de 4.2.1.5) Y de las exposiciones que me has hablado antes, ¿me podrías explicar de qué son cada una?

Respuesta: ¡Claro!

- **Proust y las artes:** Esta exposición explora la relación entre la literatura y las artes visuales. Se centra en la obra del escritor francés Marcel Proust y cómo su obra se refleja en las artes visuales de su época. La exposición muestra obras de arte de diferentes estilos y períodos, desde el siglo XIX hasta la primera mitad del siglo XX.
- **Tarek Atoui. At-Tāriq:** Esta exposición es una instalación sonora del artista Tarek Atoui, que explora la relación entre el sonido y el espacio. La instalación utiliza objetos y materiales para crear un entorno sonoro que invita a los visitantes a experimentar y reflexionar sobre la percepción del sonido.
- **Ayako Rokkaku:** Esta exposición presenta la obra de la artista japonesa Ayako Rokkaku, que se centra en la relación entre la naturaleza y la cultura. La exposición muestra obras de arte que combinan elementos naturales con objetos y materiales culturales, creando un diálogo entre la naturaleza y la humanidad.

Espero que esta información te haya ayudado a entender mejor cada exposición. ¿Te gustaría saber más sobre alguna de ellas?

4.3. Sección de “Información de Obras”

La sección *Información de obras* presenta la misma interfaz que la sección de información del museo. Por ello, se pasará directamente a la demostración de la inferencia.

4.3.1. Inferencia del modelo

4.3.1.1. Pregunta 1

Pregunta: ¿Qué es el cubismo?

Respuesta: Hola, gracias por visitar el museo Thyssen-Bornemisza. El cubismo es un movimiento artístico que surgió a principios del siglo XX, caracterizado por la fragmentación de objetos y la representación de múltiples perspectivas en una sola obra de arte. En el cubismo, los artistas buscan descomponer los objetos en sus componentes geométricos básicos, como cubos, esferas y planos, y luego reensamblarlos de manera innovadora para crear una nueva forma de ver y representar la realidad.

En nuestro museo, tenemos varias obras que ejemplifican este movimiento, como *Cubismo* de Nadeshda Udaltsova, *Botella y frutero* de Juan Gris y *Arlequín con espejo* de Pablo Picasso. Estas obras muestran cómo los artistas cubistas experimentaron con la forma y la estructura para crear composiciones innovadoras y desafiar las convenciones tradicionales de la representación artística.

¿Te gustaría saber más sobre alguna de estas obras en particular o sobre el cubismo en general?

4.3.1.2. Pregunta 2

Pregunta: Estoy buscando una obra en la que aparecía una mujer de perfil y creo que es del Renacimiento

Respuesta: ¡Hola! Me alegra poder ayudarte. Una obra de arte del Renacimiento que representa a una mujer de perfil es el *Retrato de Giovanna degli Albizzi Tornabuoni* del autor Domenico Ghirlandaio, expuesto en la sala 5. En esta pintura, Giovanna se muestra de perfil, con un busto elegante y una expresión serena. La obra es un ejemplo espléndido del retrato florentino del Quattrocento, con una atención al detalle y una idealización de las proporciones y los rasgos de la modelo. ¿Te gustaría saber más sobre esta obra o sobre otros retratos de mujeres en el museo?

4.3.1.3. Pregunta 3

Pregunta: (Continuación de la pregunta 4.3.1.2) ¿Me podrías contar alguna curiosidad de la obra?

Respuesta: ¡Claro! Una curiosidad artística relacionada con el *Retrato de Giovanna degli Albizzi Tornabuoni* es que se cree que este retrato es un ejemplo de un retrato póstumo, es decir, que fue pintado después de la muerte de Giovanna. La fecha que aparece en el cartellino del epigrama de Marcial, 1488, coincide con el año de su fallecimiento, lo que sugiere que el retrato fue encargado por su esposo, Lorenzo Tornabuoni, como un tributo a su memoria.

Además, es interesante destacar que el retrato de Giovanna se encuentra en la capilla Tornabuoni de la basílica de Santa María Novella en Florencia, donde también se encuentra un fresco de Ghirlandaio que representa a Giovanna de cuerpo entero.

Esto sugiere que el artista utilizó el mismo modelo para ambos trabajos, y que el retrato del Museo Thyssen-Bornemisza podría haber sido un estudio o un boceto para el fresco más grande.

Otra curiosidad es que el retrato de Giovanna degli Albizzi Tornabuoni es uno de los pocos retratos de mujeres del Renacimiento que se conservan en la actualidad, lo que lo hace aún más valioso y significativo en la historia del arte. ¿Te gustaría saber más sobre la vida de Giovanna o sobre otros aspectos de este retrato?

4.3.1.4. Pregunta 4

Pregunta: (Continuación de la pregunta 4.3.1.3) Me interesa saber sobre su ropa y joyas, presenta muchos ornamentos y me gustaría saber por qué llevaban tantas decoraciones en esa época

Respuesta: Hola, estoy aquí para ayudarte. En el *Retrato de Giovanna degli Albizzi Tornabuoni* de Domenico Ghirlandaio, Giovanna lleva una serie de ornamentos y joyas que reflejan su estatus social y riqueza. Estos incluyen un colgante con un rubí y perlas, un broche de oro en forma de dragón con un rubí y berilos, y un libro de oraciones.

En cuanto a por qué las mujeres del Renacimiento llevaban tantas decoraciones en su vestimenta, se debía en parte a la influencia de la moda y la cultura de la época. La ropa y los adornos eran una forma de mostrar la riqueza y el estatus social de la familia, y las mujeres de la nobleza y la burguesía se esforzaban por lucir sus mejores galas en público.

Además, la ropa y los ornamentos también tenían un significado simbólico. Por ejemplo, el rubí se asociaba con la virtud y la espiritualidad, mientras que el oro y las perlas simbolizaban la riqueza y la nobleza. Las mujeres también llevaban joyas y adornos para resaltar su belleza y atractivo, y para mostrar su posición social y marital.

En el caso de Giovanna, su ropa y joyas reflejan su estatus como miembro de la nobleza florentina y su posición como esposa de Lorenzo Tornabuoni. El retrato de Ghirlandaio la muestra como una mujer elegante y refinada, con un sentido de la moda y la cultura que era característico de la época.

4.3.1.5. Pregunta 5

Pregunta: He estado estudiando a Kandinsky y su influencia en el arte abstracto, ¿me podrías explicar un poco más y sobre todo si hay alguna obra que haya sido influenciada por el autor?

Respuesta: ¡Hola! Me alegra poder ayudarte a explorar el mundo del arte abstracto y la influencia de Kandinsky en este movimiento.

Wassily Kandinsky fue un artista ruso que jugó un papel fundamental en el desarrollo del arte abstracto. Sus teorías y obras influyeron en muchos artistas, y su legado puede verse en la obra de varios creadores que se encuentran en el Museo Thyssen-Bornemisza.

Una de las obras que podría estar influenciada por Kandinsky es *Abstracción. Resplandor I* de Georgia O'Keeffe, que se encuentra en la sala 46. Aunque O'Keeffe no

Demostración

fue una artista que se identificara directamente con el movimiento abstracto de Kandinsky, su obra comparte algunos elementos de la abstracción biomórfica que se asocian con el arte de Kandinsky.

Otra obra que podría estar influenciada por Kandinsky es *Ritmos de la tierra* de Mark Tobey, que se encuentra en la sala 46. Tobey fue un artista estadounidense que se interesó por la abstracción y la caligrafía, y su obra muestra una sensibilidad similar a la de Kandinsky en cuanto a la exploración de la forma y el color.

Sin embargo, es importante destacar que la influencia de Kandinsky se puede ver de manera más indirecta en la obra de muchos artistas que se encuentran en el museo, ya que su legado se extendió a lo largo del siglo XX y continuó influyendo en la creación artística en diversas formas y estilos.

4.3.1.6. Pregunta 6

Pregunta: Me resulta muy interesante la obra de mujer en la bañera, me recuerda a algo que he visto antes

Respuesta: La obra *Mujer en el baño* del autor Roy Lichtenstein, que creo que es a la que te refieres, evoca una sensación de familiaridad debido a su inspiración en la cultura popular y los cómics. La imagen de una mujer sonriente y relajada en el baño puede recordar a los espectadores anuncios publicitarios o escenas de películas y series de televisión que han visto antes. Sin embargo, la obra de Lichtenstein también presenta una crítica a la sociedad de consumo y la representación de la mujer en los medios de comunicación, lo que puede hacer que el espectador se detenga a reflexionar sobre la imagen y su significado. La obra también puede recordar a otras obras de arte que representan a mujeres en baños o escenas íntimas, como *El baño turco* de Ingres o *La toilette* de Boucher, que se encuentran en la misma línea de representación de la intimidad femenina.

4.3.1.7. Pregunta 7

Pregunta: (Continuación de la pregunta 4.3.1.6) Hay también una obra muy famosa, que creo haberla visto en varias referencias culturales; en la obra sale una mujer sentada en una cama leyendo un libro. Tengo entendido que se encuentra en el museo, ¿dónde está?

Respuesta: No hay una obra específica en el museo que coincida exactamente con la descripción que has proporcionado. Sin embargo, la obra *Habitación de hotel* de Edward Hopper, que se encuentra actualmente fuera de exposición, presenta a una mujer sentada en una cama, pero no está leyendo un libro, sino que sostiene un papel, probablemente un horario de trenes. Si estás interesado en ver obras que representen a mujeres en situaciones íntimas o reflexivas, podrías considerar visitar otras salas del museo, como la Sala 9, donde se encuentra *La ninfa de la fuente* de Lucas Cranach el Viejo, o la Sala 32, donde se encuentra el *Retrato de Millicent, duquesa de Sutherland* de John Singer Sargent.

4.3.1.8. Pregunta 8

Pregunta: (Continuación de la pregunta 4.3.1.7) Sí, me refería a esa exactamente, aunque no esté expuesta, ¿puedes explicarme un poco más de ella y decirme si hay alguna obra del mismo autor o similar?

Respuesta: ¡Hola! Me alegra poder ayudarte a conocer más sobre la obra *Habitación de hotel* de Edward Hopper. Aunque no está expuesta en el museo Thyssen-Bornemisza, puedo proporcionarte información sobre ella.

Habitación de hotel es una obra maestra de Edward Hopper, pintada en 1931. La pintura muestra a una mujer sentada en el borde de una cama en una habitación de hotel, sosteniendo un papel, probablemente un horario de trenes. La escena es característica del estilo de Hopper, con una iluminación cenital fría y una atmósfera de soledad y melancolía.

La obra es una representación de la alienación y la soledad en la vida moderna, temas que Hopper exploró en muchas de sus obras. La habitación de hotel, con su decoración impersonal y su falta de conexión con el exterior, se convierte en un símbolo de la desconexión y la soledad de la mujer.

Aunque *Habitación de hotel* no esté expuesta en el museo, hay otras obras de Edward Hopper que podrían ser de interés para ti. Por ejemplo, podrías buscar obras que también exploren temas de soledad y alienación, como *Nighthawks* (1942) o *Chop Suey* (1929). Estas obras también muestran la habilidad de Hopper para capturar la esencia de la vida moderna y la condición humana.

Si estás interesado en ver obras similares, podrías considerar visitar la sección de arte americano del siglo XX en el museo, donde podrías encontrar obras de otros artistas que también exploraron temas de soledad y alienación, como Grant Wood o John Sloan.

¿Quieres saber más sobre la vida y la obra de Edward Hopper, o sobre otras obras que podrían ser de interés para ti?

4.4. Sección de “Recorridos temáticos”

La sección de *Recorridos temáticos* presenta una interfaz diferenciada del resto de partes de Museion como consecuencia de los requisitos funcionales y de diseño expuestos previamente en el capítulo dedicado al desarrollo de la plataforma.

Al acceder a esta sección, el usuario es recibido con una pantalla que muestra los distintos recorridos disponibles, cada uno acompañado de una breve descripción. En la parte inferior de la interfaz se proporciona un mecanismo de selección mediante el cual el usuario puede indicar el recorrido temático que desea iniciar. La figura 4.2 ilustra la disposición visual de esta interfaz.

Una vez seleccionado el recorrido, comienza la descripción extensa del mismo. Esta descripción ayuda al visitante a entender el contexto sobre el cual se van a explicar las obras. La interfaz se presenta tal y como se ilustra en la Figura 4.3.

Una vez se ha explicado el contexto del recorrido, se presenta un tutorial al visitante. La interfaz que se presenta en la Figura 4.4 y en la Figura 4.5 se repite obra a obra.

Una vez seleccionado el recorrido, se inicia la explicación de la primera obra en el contexto del itinerario correspondiente. Para ello, el *chatbot* genera una serie de mensajes que contienen la descripción detallada de la pieza. Al concluir dicha explicación, la interfaz presenta al usuario un campo de entrada de texto junto con la opción de continuar con el recorrido mediante un botón.

En caso de que el usuario desee formular alguna pregunta relacionada con la obra, puede hacerlo a través del campo de texto habilitado para ello. Si, por el contrario, op-

 ¡De acuerdo! El Museo Nacional Thyssen-Bonersmiza dispone de varios recorridos temáticos. Algunos de ellos son los siguientes.



Amor diverso

Una mirada distinta, basada en género, identidad y orientación sexual.

Este recorrido está compuesto por 16 obras.



La cultura del vino

Los diferentes aspectos de la historia del vino plasmados en diversos cuadros

Este recorrido está compuesto por 19 obras.



La moda

Un viaje por la evolución de la indumentaria desde el siglo XIV hasta nuestros días.

Este recorrido está compuesto por 15 obras.

 Puedes seleccionar alguno de los anteriores o consultar más recorridos.

Amor diverso

La cultura del vino

La moda

Más recorridos

Figura 4.2: Interfaz de elección de recorrido temático

ta por no realizar ninguna consulta, tiene la posibilidad de continuar con la siguiente obra pulsando el botón correspondiente, lo que permite avanzar en el recorrido.

4.4.1. Inferencia del modelo

4.4.1.1. Pregunta 1

En el contexto de la obra *La Virgen y el Niño con un racimo de uvas* de Lucas Cranach el Viejo.

Pregunta: Hablas del Renacimiento alemán, ¿qué diferencia hay entre el Renacimiento alemán y el italiano?

Respuesta: ¡Hola! Me alegra que estés interesado en conocer más sobre el Renacimiento alemán y su diferencia con el Renacimiento italiano.

El Renacimiento alemán, como se ve en obras como *La Virgen y el Niño con un racimo de uvas*, de Lucas Cranach el Viejo, se diferencia del Renacimiento italiano en varios aspectos. Una de las principales diferencias es el enfoque en la representación de la figura humana. Mientras que el Renacimiento italiano se centró en la creación de cuerpos idealizados y proporcionalmente perfectos, el Renacimiento alemán tendió a representar figuras más naturalistas y menos idealizadas.

4.4. Sección de “Recorridos temáticos”



Figura 4.3: Interfaz de explicación del recorrido

En el caso de Cranach el Viejo, su estilo se caracteriza por la creación de un prototipo femenino inconfundible en el Renacimiento alemán, que se diferencia del ideal de belleza italiano. Sus figuras femeninas suelen tener rostros más redondos y menos clásicos que las de los artistas italianos.

Además, el Renacimiento alemán también se distingue por su enfoque en la narrativa y la simbología, en lugar de la simple representación de la forma y la belleza. En *La Virgen y el Niño con un racimo de uvas*, por ejemplo, el racimo de uvas es un símbolo de la Eucaristía y del papel de redentor de Jesús, lo que refleja la importancia de la narrativa y la simbología en el arte alemán de la época.

En resumen, el Renacimiento alemán se diferencia del Renacimiento italiano en su enfoque en la representación de la figura humana, su estilo y su énfasis en la narrativa y la simbología. ¿Te gustaría saber más sobre el Renacimiento alemán o sobre la obra de Cranach el Viejo en particular?

4.4.1.2. Pregunta 2

En el contexto de la obra *La Virgen y el Niño con un racimo de uvas* de Lucas Cranach el Viejo y continuación de la pregunta 4.4.1.1

 A partir de ahora iré guiándole por el museo durante el recorrido. Antes de describir la obra le diré a que sala debe dirigirse y cuando esté en ella pulse el botón "Pulse para continuar" como el que aparece ahora mismo.

Pulse para continuar

Figura 4.4: Interfaz de explicación del recorrido

 Continuar con la explicación de la obra.

 ¡Perfecto! Parece que entiendes la dinámica. Cuando pulses continuar procederé a describir la siguiente obra del recorrido, una vez terminada la explicación podrá preguntarme cualquier duda acerca de la obra o pulsar el botón "Continuar con la siguiente obra" para continuar.
¡Probémoslo!

Continuar con la siguiente obra

Figura 4.5: Interfaz de explicación del recorrido

Pregunta: ¿Por qué aparece en un paisaje tan europeo si la imagen que representa es una escena bíblica y por tanto debería ser en Oriente?

Respuesta: ¡Hola de nuevo! Me alegra que estés interesado en conocer más sobre la representación de escenas bíblicas en el arte europeo.

La razón por la que se representa una escena bíblica como *La Virgen y el Niño con un racimo de uvas*, de Lucas Cranach el Viejo en un entorno europeo, a pesar de que la escena en sí debería estar ambientada en Oriente, se debe a varias razones.

En primer lugar, durante el Renacimiento, los artistas europeos no tenían una concepción muy clara de la geografía y la cultura del Oriente Medio, donde se desarrollan la mayoría de las escenas bíblicas. Por lo tanto, era común que los artistas representaran estas escenas en un entorno más familiar para ellos, es decir, en un entorno europeo.

En segundo lugar, la representación de escenas bíblicas en un entorno europeo también se debía a la intención de hacer que la historia bíblica fuera más accesible y comprensible para el público europeo. Al representar las escenas en un entorno familiar, los artistas podían ayudar a los espectadores a conectarse emocionalmente con la historia y a entender mejor su significado.

En el caso de *La Virgen y el Niño con un racimo de uvas*, de Cranach el Viejo, la representación de la escena en un entorno europeo también se debe a la influencia del arte medieval y gótico, que solía representar escenas bíblicas en un entorno más fantástico y menos histórico. Cranach el Viejo, como muchos otros artistas de su época, se inspiró en estas tradiciones artísticas y las incorporó en su propia obra.

Finalmente, es importante destacar que la representación de escenas bíblicas en un entorno europeo también se debía a la intención de transmitir un mensaje teológico y moral. Al representar las escenas en un entorno familiar, los artistas podían destacar

la universalidad del mensaje bíblico y su relevancia para la vida cotidiana de los espectadores.

En resumen, la representación de escenas bíblicas en un entorno europeo, como en *La Virgen y el Niño con un racimo de uvas*, de Cranach el Viejo, se debía a una combinación de factores, incluyendo la falta de conocimiento sobre la geografía y la cultura del Oriente Medio, la intención de hacer que la historia bíblica fuera más accesible y comprensible, la influencia del arte medieval y gótico, y la intención de transmitir un mensaje teológico y moral. ¿Te gustaría saber más sobre la representación de escenas bíblicas en el arte europeo?

4.4.1.3. Pregunta 3

En el contexto de la obra *Retrato de Matthäus Schwarz* del autor Christoph Amberger.

Pregunta: ¿Y por qué era tan famoso?

Respuesta: ¡Hola! Me alegra poder ayudarte con tu pregunta.

Matthäus Schwarz era un personaje destacado en la sociedad de Augsburgo en el siglo XVI, y su fama se debía a varias razones. Era un contable de la familia Fugger, una de las familias más poderosas y ricas de Alemania en ese momento. Además, Matthäus Schwarz era un hombre culto y escribió varios tratados sobre contabilidad, lo que lo convirtió en una figura respetada en su campo.

Sin embargo, su fama también se debía a su libro *Trachtenbuch*, un manuscrito que contiene 137 ilustraciones de los trajes más importantes que lució a lo largo de su vida, acompañadas de anotaciones y descripciones. Este libro es considerado un valioso documento sobre la moda masculina de la época y ofrece una visión única de la vida y la cultura de la nobleza y la burguesía en el Renacimiento alemán.

En el contexto de la obra *Retrato de Matthäus Schwarz* del autor Christoph Amberger, la fama de Matthäus Schwarz se refleja en la forma en que el artista lo representa. El retrato muestra a Matthäus Schwarz como un hombre próspero y culto, rodeado de símbolos de su profesión y su estatus social. La obra es un ejemplo de la habilidad de Christoph Amberger para capturar la esencia y la personalidad de sus sujetos, y su importancia radica en la forma en que refleja la cultura y la sociedad de la época.

Capítulo 5

Evaluación

5.1. Evaluación de la capacidad agencial del sistema multiagente de la sección “Consultar obras”

En esta sección se procederá a evaluar la eficacia de la capacidad agencial del sistema multiagente implementado en la sección *Consultar obras*. La capacidad agencial hace referencia a la capacidad del sistema para organizar y dirigir de manera autónoma el flujo de trabajo del sistema multiagente.

Más concretamente, esta capacidad implica que el sistema es capaz de: (i) interpretar la *query* del visitante, (ii) clasificarla en función del tipo de tarea subyacente, y (iii) seleccionar el agente más adecuado para ejecutar dicha tarea de forma eficiente.

5.1.1. Método de evaluación utilizado

Para evaluar la eficacia de la capacidad agencial del sistema multiagente, se ha adoptado un enfoque basado en clasificación supervisada. El procedimiento seguido consiste en utilizar un conjunto de datos compuesto por consultas previamente clasificadas, es decir, cada entrada del conjunto de evaluación está etiquetada con la clase correcta que representa la tarea correspondiente.

El proceso de evaluación se ha llevado a cabo en las siguientes etapas:

1. **Selección del conjunto de evaluación:** Se ha construido un conjunto de datos compuesto por una colección representativa de *queries* reales o simuladas, cada una de las cuales ha sido previamente anotada con la clase o categoría que le corresponde según el diseño funcional del sistema.
2. **Inferencia del modelo:** Para cada una de las consultas del conjunto, se ha ejecutado el sistema multiagente, permitiendo que este determine de forma autónoma la clasificación correspondiente a la tarea. En este contexto, la clasificación equivale a la elección del agente más adecuado para llevar a cabo dicha tarea.
3. **Comparación con las etiquetas reales:** Una vez obtenida la clasificación propuesta por el sistema, se ha comparado con la clase real asociada a cada consulta. Esta comparación permite cuantificar el grado de acierto del sistema y, en consecuencia, valorar su capacidad para interpretar correctamente las peticiones y asignarles el agente adecuado.

5.1. Evaluación de la capacidad agencial del sistema multiagente de la sección “Consultar obras”

Cada una de las evaluaciones ha sido aplicada de manera independiente a dos modelos de lenguaje distintos: `llama-3.3-70b-versatile` y `llama-3.1-8b-instant`, ambos accesibles a través de la API proporcionada por Groq.

A priori, el modelo `llama-3.3-70b-versatile`, de mayor tamaño, está diseñado para tareas que requieren un mayor grado de comprensión contextual y toma de decisiones complejas, mientras que el modelo `llama-3.1-8b-instant` prioriza la rapidez en la inferencia, siendo más adecuado para sistemas que requieren respuestas en tiempo real. La comparación entre ambos modelos permite obtener una visión más completa sobre el equilibrio entre precisión y eficiencia computacional en el contexto del sistema multiagente desarrollado.

Este enfoque permite medir de forma objetiva el rendimiento del modelo, proporcionando una base sólida para el cálculo de métricas estándar en tareas de clasificación, como la exactitud (*accuracy*), la precisión, el *recall* y *F1-score*, si así se requiere.

La adopción de este método garantiza una evaluación sistemática y reproducible del componente de decisión del sistema agencial, facilitando tanto la identificación de posibles errores como la mejora iterativa del modelo.

5.1.2. Datos utilizados

Para llevar a cabo la evaluación del sistema multiagente, se ha generado un conjunto de datos sintético compuesto por un total de 80 consultas representativas que podrían ser formuladas por visitantes reales en el contexto de la sección *Consultar obras*. Estas preguntas han sido diseñadas manualmente para cubrir un espectro amplio de posibles intenciones y escenarios de interacción, asegurando una diversidad semántica y funcional que permita evaluar adecuadamente la capacidad de clasificación del sistema.

Cada consulta del conjunto ha sido anotada siguiendo dos esquemas de clasificación distintos, con el fin de contrastar el comportamiento del modelo bajo diferentes niveles de granularidad:

- **Clasificación en 6 clases:** Las consultas se han categorizado según las siguientes etiquetas: `description`, `location`, `activities`, `tours`, `information` y `n_related`. Este esquema permite una segmentación más detallada de las intenciones del usuario y abre las puertas a añadir algunas de las funcionalidades de la sección *Información del museo*.
- **Clasificación en 4 clases:** Con el objetivo de simplificar el espacio de decisión, se ha aplicado una segunda taxonomía reducida que agrupa las consultas bajo las etiquetas: `description`, `location`, `n_related` y `other`. En este esquema, las categorías `activities` y `tours` se incorporan dentro de `other`, al considerarse preguntas que se encuentran fuera del alcance del sistema.

5.1.3. Resultados

Los resultados obtenidos para cada evaluación se analizarán en esta sección.

El Cuadro 5.1.3 muestra la matriz de confusión correspondiente a la evaluación del modelo `llama-3.3-70b-versatile` clasificado en cuatro clases: `description`, `location`, `other` y `n_related`. A partir de ella, se puede observar un buen rendimiento general del modelo, respaldado por las métricas obtenidas: un **accuracy** del **80%**, una **precision** del **85%**, un **recall** del **80%** y un **F1 Score** del **81%**.

Evaluación

		Predicción			
		description	location	other	n_related
Real	description	29	2	0	2
	location	0	4	1	0
	other	6	5	25	0
	n_related	0	0	0	5

Cuadro 5.1: Matriz de confusión para la evaluación del modelo *llama-3.3-70b-versatile* con cuatro clases

En cuanto a la distribución de los aciertos y errores:

- La clase *description* presenta un alto nivel de acierto (29 de 33 instancias), aunque muestra cierta confusión con las clases *location* (2 errores) y *n_related* (2 errores).
- La clase *location* se clasifica correctamente en la mayoría de los casos (4 de 5), pero se confunde una vez con *other*.
- La clase *other* muestra un rendimiento más bajo, con 6 instancias mal clasificadas como *description* y 5 como *location*, aunque mantiene un núcleo importante de aciertos (25 de 36).
- Finalmente, la clase *n_related* se clasifica de forma precisa y sin ambigüedad (5 aciertos de 5).

Estos resultados sugieren que el modelo tiene una capacidad sólida para identificar correctamente las clases más relevantes y distintivas, como *description* y *location*. Para el caso de *other* los errores no son especialmente significativos, ya que en caso de ejecutarse los agentes descriptivos y de localización y no encontrar una respuesta satisfactoria, el modelo diría que no tiene respuesta para esa pregunta, lo que se considera también una respuesta satisfactoria.

		Predicción			
		description	location	other	n_related
Real	description	30	2	0	1
	location	1	4	0	0
	other	6	4	25	1
	n_related	1	0	0	4

Cuadro 5.2: Matriz de confusión para la evaluación del modelo *llama-3.1-8b-instant* con cuatro clases.

El Cuadro 5.2 presenta la matriz de confusión obtenida tras la evaluación del modelo *llama-3.3-70b-versatile* con un esquema de clasificación en cuatro clases: *description*, *location*, *other* y *n_related*. A partir de ella, se derivan las siguientes métricas: una **accuracy** del **80%**, una **precision** del **85%**, un **recall** del **80%** y un **F1 Score** del **81%**, lo que refleja un rendimiento global positivo del sistema.

Analizando los errores de clasificación más significativos:

- La clase *description* fue correctamente identificada en 30 de las 33 ocasiones, cometiéndose 2 errores hacia *location* y 1 hacia *n_related*.

5.1. Evaluación de la capacidad agencial del sistema multiagente de la sección “Consultar obras”

- La clase *location* obtuvo un desempeño casi perfecto, con 4 aciertos sobre 5 instancias y únicamente 1 error hacia *description*.
- La clase *other*, con 36 instancias, fue correctamente identificada en 25 casos. Sin embargo, se produjeron 6 clasificaciones incorrectas como *description*, 4 como *location* y 1 como *n_related*, lo que evidencia una mayor confusión en esta categoría.
- Finalmente, *n_related* fue correctamente clasificada en 4 de 5 ocasiones, con 1 error hacia *description*.

En conjunto, se presentan resultados muy similares al caso anterior.

		Predicción					
		description	location	activities	tours	information	n_related
Real	description	30	2	0	0	0	1
	location	1	4	0	0	0	0
	activities	6	3	13	2	0	1
	tours	0	0	2	4	0	0
	information	0	0	0	0	4	0
	n_related	0	0	0	0	0	5

Cuadro 5.3: Matriz de confusión para la evaluación del modelo *llama-3.3-70b-versatile* con seis clases.

El Cuadro 5.3 muestra la matriz de confusión obtenida al evaluar el modelo *llama-3.3-70b-versatile* sobre un conjunto de datos clasificado en seis categorías: *description*, *location*, *activities*, *tours*, *information* y *n_related*. El modelo alcanzó un **accuracy** del **79%**, una **precision** del **83%**, un **recall** del **79%** y un **F1 Score** del **79%**.

En detalle:

- La clase *description* muestra un alto nivel de acierto, con 30 instancias clasificadas correctamente de un total de 33, y solo tres errores (2 hacia *location* y 1 hacia *n_related*).
- *Location* presenta un buen desempeño con 4 aciertos sobre 5 instancias, cometiendo un único error hacia *description*.
- La clase *activities* evidencia mayores dificultades para el modelo: de 25 instancias, 13 fueron correctamente clasificadas, mientras que 6 se confundieron con *description*, 3 con *location*, 2 con *tours* y 1 con *n_related*.
- En el caso de *tours*, se observaron 4 aciertos y 2 errores (ambos clasificados como *activities*), lo que podría indicar una proximidad entre ambas clases.
- La clase *information* fue perfectamente clasificada, sin errores (4 de 4).
- Finalmente, *n_related* contiene 6 instancias, de las cuales 5 fueron correctamente clasificadas y una fue confundida con *activities*.

En general, el modelo logra un rendimiento aceptable en categorías claramente definidas como *description*, *location*, *information* y *n_related*, mientras que las categorías *activities* y *tours* presentan un mayor grado de confusión. Este comportamiento podría explicarse por la ambigüedad y solapamiento en las temáticas.

		Predicción					
		description	location	activities	tours	information	n_related
Real	description	31	1	0	0	0	1
	location	1	4	0	0	0	0
	activities	3	3	17	0	1	1
	tours	0	0	4	2	0	0
	information	0	0	0	0	4	0
	n_related	0	0	0	0	0	4

Cuadro 5.4: Matriz de confusión para la evaluación del modelo *llama-3.1-8b-instant* con seis clases.

Por último, el Cuadro 5.4 presenta la matriz de confusión correspondiente a la evaluación del modelo *llama-3.1-8b-instant* con una clasificación en seis clases: *description*, *location*, *activities*, *tours*, *information* y *n_related*. Las métricas obtenidas indican un rendimiento aceptable, con un **accuracy** del **77%**, una **precision** del **80%**, un **recall** del **77%** y un **F1 Score** del **76%**.

A continuación se detalla el comportamiento del modelo para cada clase:

- La clase *description* presenta un muy buen rendimiento, con 31 aciertos sobre 33 instancias. Los errores son mínimos y se reparten entre las clases *location* y *n_related*.
- *Location* también muestra un desempeño adecuado, con 4 clasificaciones correctas sobre 5, y un único error hacia *description*.
- La clase *activities* ofrece un rendimiento moderado. De las 25 instancias, 17 fueron correctamente clasificadas, mientras que las restantes se distribuyeron en errores hacia *description* (3), *location* (3), *information* (1) y *n_related* (1).
- En la clase *tours*, se observan 2 aciertos sobre 6 instancias, con 4 errores que fueron clasificados como *activities*, lo que sugiere un solapamiento semántico entre ambas clases.
- La categoría *information* fue correctamente identificada en su totalidad (4 aciertos), sin errores de clasificación.
- Finalmente, la clase *n_related* muestra un buen desempeño con 4 aciertos sobre 5. Cabe destacar que el error cometido en la clasificación se debe a que la respuesta obtenida por el LLM no tenía el formato correcto.

Finalmente, presenta los mismos errores que para el caso anterior pero más acentuados.

5.1.3.1. Conclusiones de los resultados

Como se ha podido evidenciar en los resultados obtenidos, la granularización de las clases produce una disminución en el rendimiento. Además, dicha granularización implica la necesidad de desarrollar agentes adicionales especializados para el tratamiento de categorías específicas, lo que conlleva un incremento en el consumo de recursos, especialmente en términos de uso de *tokens*. Por estas razones, se ha op-

tado por adoptar una clasificación más reducida, compuesta por únicamente cuatro clases, tal como se ha descrito previamente en el Capítulo 3.

Asimismo, los resultados obtenidos para los dos modelos analizados —*llama-3.3-70b-versatile* y *llama-3.1-8b-instant*— son comparables en términos de métricas de rendimiento. No obstante, considerando tanto el coste económico asociado al uso de cada modelo como las restricciones en las tasas de *tokens*, se ha decidido emplear el modelo de menor tamaño, *llama-3.1-8b-instant*.

5.2. Evaluación del agente conversacional de descripciones

Para evaluar la calidad de todos los elementos involucrados durante la generación de respuestas en el agente de descripciones, es decir, todos los elementos intermedios a la hora de realizar el RAG descritos en el capítulo 3, se ha llevado a cabo una evaluación que permite valorar dicha herramienta basándose en métricas concretas de evaluación.

5.2.1. Métricas de evaluación utilizadas

En cuanto a métricas de evaluación para sistemas basados en arquitecturas RAG, se presentan varias formas de cuantificar la calidad de las respuestas obtenidas en función de las características tanto de la respuesta como del contexto aportado [48]. Con el fin de facilitar el proceso de evaluación, se han tomado como referencia las métricas presentadas en la página de documentación del framework utilizado para desarrollar el proyecto. La documentación, así como una explicación de cada una de las métricas, puede ser consultada en la página web de LangChain¹.

- **Relevance:** evalúa qué tan bien la respuesta generada por el modelo responde a la pregunta o instrucción dada por el usuario. Para ello se comparará la respuesta dada con la pregunta del usuario mediante un *LLM-as-judge* para evaluar la relevancia, utilidad, entre otras.
- **Groundedness:** mide hasta qué punto la respuesta del modelo está basada o justificada en las fuentes proporcionadas, es decir, si se apoya en el contexto recuperado. Para ello se comparará la respuesta con el contexto obtenido mediante un *LLM-as-judge* para evaluar la fidelidad, alucinaciones, entre otras.
- **Retrieval relevance:** evalúa qué tan bien los documentos o fragmentos recuperados por el RAG están relacionados con la pregunta del usuario. Al igual que las anteriores utiliza un *LLM-as-judge* para evaluar la relevancia.

5.2.2. Metodología de evaluación

Para evaluar la capacidad de respuesta del modelo, se han utilizado las tres métricas descritas anteriormente, empleando un enfoque *LLM-as-a-judge*. Esta técnica permite delegar la evaluación a modelos de lenguaje, agilizando significativamente el proceso y reduciendo la necesidad de intervención humana [48, 49].

La elección de este método se debe, principalmente, a la falta de información y conocimiento especializado para determinar con certeza si una respuesta es correcta o no. Dado que no se cuenta con un experto en la materia, no sería posible resolver

¹<https://docs.smith.langchain.com/evaluation/tutorials/rag>

Evaluación

estas dudas de manera manual. Por este motivo, la evaluación se ha centrado exclusivamente en la calidad de la respuesta con respecto al contexto proporcionado y a la pregunta realizada, sin considerar la veracidad absoluta de la respuesta por la ausencia de un criterio experto.

Para llevar a cabo la evaluación, se ha utilizado el modelo de OpenAI `gpt-4`. El motivo de su elección se debe a su notable superioridad en comparación con otros modelos disponibles de forma gratuita para la realización de este trabajo. No obstante, debido al alto coste asociado al uso de este modelo, el número de pruebas realizadas tuvo que ser limitado para mantenerse dentro del alcance del proyecto.

Esta superioridad se refleja en la Figura 5.1, donde se observa que `gpt-4` supera a otros modelos de la misma compañía en varias de las temáticas analizadas.

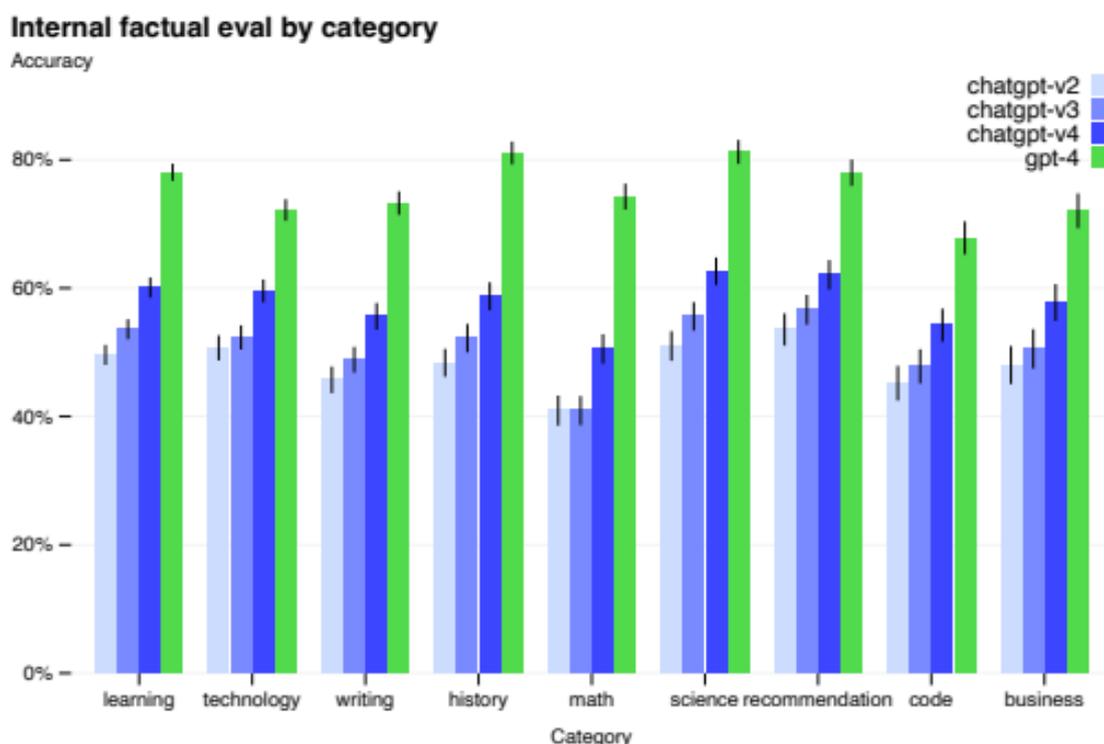


Figura 5.1: Desempeño de los modelos de OpenAI en distintas temáticas [50].

Además, en la Figura 5.2, se aprecia que el modelo de OpenAI se sitúa por encima de sus competidores, siendo únicamente superado en algunas evaluaciones relevantes para este proyecto—como la capacidad de razonamiento y la ventana de contexto—por Llama 3 405B. Sin embargo, debido a su gran tamaño, disponibilidad y precio, este modelo resulta inaccesible para el presente trabajo.

Para la evaluación se han utilizado los *prompts* propuestos en la documentación de la herramienta LangChain que ha sido utilizada para la realización del código. La evaluación consiste en otorgar una puntuación de 1 si se considera que la respuesta cumple con lo esperado en cuanto a las características evaluadas o un 0 si se considera que no se cumplen.

Por último, se ha utilizado la herramienta LangSmith, que permite evaluar todos los pasos durante la inferencia.

5.2. Evaluación del agente conversacional de descripciones

Category	Benchmark	Llama 3 8B	Gemma 2 9B	Mistral 7B	Llama 3 70B	Mixtral 8x22B	GPT 3.5 Turbo	Llama 3 405B	Nemotron 4 340B	GPT-4 <small>o1-mini</small>	GPT-4o	Claude 3.5 Sonnet
General	MMLU <small>(5-shot)</small>	69.4	72.3	61.1	83.6	76.9	70.7	87.3	82.6	85.1	89.1	89.9
	MMLU <small>(0-shot, CoT)</small>	73.0	72.3 ^Δ	60.5	86.0	79.9	69.8	88.6	78.7 ^Δ	85.4	88.7	88.3
	MMLU-Pro <small>(5-shot, CoT)</small>	48.3	–	36.9	66.4	56.3	49.2	73.3	62.7	64.8	74.0	77.0
	IFEval	80.4	73.6	57.6	87.5	72.7	69.9	88.6	85.1	84.3	85.6	88.0
Code	HumanEval <small>(0-shot)</small>	72.6	54.3	40.2	80.5	75.6	68.0	89.0	73.2	86.6	90.2	92.0
	MBPP EvalPlus <small>(0-shot)</small>	72.8	71.7	49.5	86.0	78.6	82.0	88.6	72.8	83.6	87.8	90.5
Math	GSM8K <small>(8-shot, CoT)</small>	84.5	76.7	53.2	95.1	88.2	81.6	96.8	92.3 [∇]	94.2	96.1	96.4 [∇]
	MATH <small>(0-shot, CoT)</small>	51.9	44.3	13.0	68.0	54.1	43.1	73.8	41.1	64.5	76.6	71.1
Reasoning	ARC Challenge <small>(0-shot)</small>	83.4	87.6	74.2	94.8	88.7	83.7	96.9	94.6	96.4	96.7	96.7
	GPQA <small>(0-shot, CoT)</small>	32.8	–	28.8	46.7	33.3	30.8	51.1	–	41.4	53.6	59.4
Tool use	BFCL	76.1	–	60.4	84.8	–	85.9	88.5	86.5	88.3	80.5	90.2
	Nexus	38.5	30.0	24.7	56.7	48.5	37.2	58.7	–	50.3	56.1	45.7
Long context	ZeroSCROLLS/QuALITY	81.0	–	–	90.5	–	–	95.2	–	95.2	90.5	90.5
	InfiniteBench/En.MC	65.1	–	–	78.2	–	–	83.4	–	72.1	82.5	–
	NIH/Multi-needle	98.8	–	–	97.5	–	–	98.1	–	100.0	100.0	90.8
Multilingual	MGSM <small>(0-shot, CoT)</small>	68.9	53.2	29.9	86.9	71.1	51.4	91.6	–	85.9	90.5	91.6

Figura 5.2: Evaluación de los modelos Llama, GPT, Gemma, Mistral y Claude [51].

5.2.3. Datos utilizados

Para la realización de las pruebas, se ha creado un *dataset* sintético con preguntas relacionadas con el museo, divididas en las siguientes temáticas:

- **Preguntas generales:** este *dataset* lo componen un total de 12 ejemplos, los cuales hacen referencia a preguntas de tipo general. Estas preguntas de tipo general son preguntas acerca de movimientos artísticos, temáticas dentro del arte, entre otras. Algunos ejemplos son: “¿Qué es el arte Pop?”, “¿Qué corrientes hay dentro del expresionismo?”, “¿Dónde surge el Rococó?”.
- **Preguntas sobre descripciones:** este *dataset* lo componen un total de 32 ejemplos, los cuales hacen referencia a preguntas concretas acerca de descripciones de obras o autores. Algunos ejemplos son: “Cuéntame la vida de Van Gogh”, “Explicame la obra retrato de Millicent”, “¿Qué caracteriza a Kandinsky?”, “Explícame las etapas artísticas de Manet”.
- **Preguntas sobre contextos históricos:** este *dataset* lo componen un total de 21 ejemplos, los cuales hacen referencia a contextos históricos en los que se desarrolla la carrera de un artista o se realiza una obra concreta. Algunos ejemplos son: “¿En qué contexto histórico se pintó la obra hombre rojo con bigote?”, “¿En qué contexto histórico desarrolla su carrera Mondrian?”.
- **Preguntas sobre detalles específicos:** este *dataset* lo componen un total de 24 ejemplos, los cuales hacen referencia a preguntas muy concretas respecto a obras o elementos dentro de obras. Algunos ejemplos son: “¿Qué representa la espada de Santa Catalina en la obra de Caravaggio?”, “¿Qué quieren decir los escritos de la obra retrato de Giovanna?”.
- **Preguntas sobre detalles generales:** este *dataset* lo componen un total de 27 ejemplos, los cuales hacen referencia a preguntas concretas sobre obras, pero dejan al modelo de lenguaje determinar qué tipo de respuesta quieren dar, ya que no se especifica nada concreto en la pregunta. Algunos ejemplos son:

Evaluación

“Cuéntame alguna curiosidad sobre el Retrato de Giovanna”, “¿Tiene algo que ver Warhaol con la obra Mujer en el Baño?”

Los motivos de dividir las preguntas en *datasets* divididos por temáticas responden a las siguientes necesidades. La primera, aislar casos concretos y observar los puntos fuertes y débiles del *chatbot* dividiendo por temáticas. La segunda, evitar problemas con tasas de *tokens* para eludir los límites de las tarifas gratuitas de los modelos utilizados.

5.2.4. Resultados

Los resultados obtenidos se pueden observar en el Cuadro 5.5.

<i>Dataset</i>	<i>Relevance</i>	<i>Groundedness</i>	<i>Retrieval Relevance</i>
Preguntas generales	0,50	0,92	1,00
Descripciones	0,78	0,94	1,00
Contexto histórico	0,86	0,90	0,90
Detalles Específicos	0,25	1,00	1,00
Detalles generales	0,96	0,96	1,00
Total	0,670	0,944	0,980

Cuadro 5.5: Evaluación de la calidad de las respuestas en diferentes categorías de preguntas.

Los resultados obtenidos son correctos y aceptables en todas las métricas y conjuntos de datos analizados. Es importante destacar la tasa de *groundedness* observada en los conjuntos de datos correspondientes a detalles específicos y preguntas generales.

En estos dos casos, se ha registrado un resultado particularmente bajo, lo que sugiere la posibilidad de la aparición de alucinaciones potenciales en el modelo. Para comprender mejor este resultado, es necesario llevar a cabo un análisis más detallado de la casuística específica que podría haber influido en dichos valores. Tras evaluar al detalle los casos en los que se detectó que el contexto atribuido no tenía relación con la pregunta realizada por el usuario, es decir, aquellos en los que la métrica *groundedness* no fue evaluada con éxito, se obtienen las siguientes conclusiones.

Para las preguntas fallidas en el dataset de preguntas generales, las métricas de *relevance* y *retrieval relevance* son evaluadas con éxito. Teniendo en cuenta que la métrica de *groundedness* resultó fallida, pero las otras dos métricas evaluadas no, se da a entender que la respuesta sí que responde a la pregunta realizada y sí que está basada en el contexto dado. Al revisar al detalle los casos, se concluye que los documentos aportados como contexto, pese a tratar el tema central de la pregunta, lo hacen de una forma muy específica y centrada en el contexto de la obra a la que pertenecen. Igualmente, la respuesta generada se basa en los documentos aportados como contexto, por lo que se asume que no ha habido alucinaciones.

Para las preguntas fallidas en el dataset de detalles específicos, vuelve a repetirse el caso anterior.

Capítulo 6

Resultados y conclusiones

6.1. Resultados

Los resultados obtenidos se alinean con los objetivos definidos al inicio del trabajo. Este Trabajo Fin de Máster ha culminado con el desarrollo de una herramienta funcional que pone en valor el potencial de la inteligencia artificial en el ámbito cultural, demostrando su capacidad para transformar la experiencia del visitante.

La solución propuesta no solo confirma la viabilidad técnica del uso de modelos de lenguaje en este contexto, sino que también pone en valor su impacto en términos de accesibilidad, personalización y eficiencia operativa.

En consecuencia, se puede afirmar que los resultados obtenidos han sido satisfactorios, en la medida en que han permitido cumplir con los objetivos inicialmente propuestos. Estos resultados no solo evidencian la validez del enfoque metodológico adoptado, sino que también demuestran la eficacia de las estrategias implementadas para abordar las distintas fases del proyecto.

6.2. Conclusiones personales

Siguiendo con las líneas de lo expuesto a lo largo de este capítulo, se han alcanzado los resultados que se esperaban. Es importante señalar que el desarrollo de este trabajo se ha visto notablemente condicionado por la constante evolución del sector. Dado el impacto que este ámbito tiene en la sociedad actual, numerosos modelos han sido actualizados, presentándose nuevas versiones e incluso nuevas propuestas por parte de otros competidores. Este dinamismo no solo ha afectado a los modelos en sí, sino también a las herramientas necesarias para su implementación.

Como consecuencia, algunas de las herramientas y modelos empleados no han sido de última generación, y se han quedado fuera algunos de los modelos más recientes como la serie “o” de OpenAI o DeepSeek, modelos de razonamiento que tienen mejores resultados a la hora de tomar decisiones, y por tanto, hubiesen sido mejores para la implementación del sistema multiagente. Asimismo, este desarrollo ha afectado también a herramientas para la creación de agentes, como CrewAI o AutoGen (Microsoft), que también han quedado fuera del proyecto.

Otras limitaciones significativas han sido la capacidad de cómputo y los costes asociados al uso de los modelos. Es ampliamente reconocido que tanto el uso como el

entrenamiento de este tipo de modelos conllevan elevados requerimientos computacionales. Por esta razón, el desarrollo del proyecto se ha visto restringido en este aspecto. A pesar de haberse empleado modelos de pago, como GPT-4, o haber recurrido a los planes de pago de la plataforma Groq para garantizar el funcionamiento fluido de las demostraciones —dado que el consumo de *tokens* superaba los límites de las tarifas gratuitas—, el uso de dichos recursos se ha limitado considerablemente al haber sido financiado íntegramente de forma personal.

En este contexto, es razonable pensar que podrían haberse obtenido resultados más avanzados mediante el uso de modelos más potentes o mediante técnicas de *fine-tuning*. Sin embargo, la falta de capacidad computacional, tanto para el entrenamiento como para la ejecución sin latencia, así como la carencia de datos de entrenamiento adecuados —en particular, conversaciones validadas en términos de efectividad—, ha imposibilitado la viabilidad de estas estrategias. Cabe mencionar que se consideró dicha opción en las fases iniciales del proyecto, aunque finalmente se descartó por las razones expuestas.

En definitiva, aunque el sistema desarrollado no constituye una solución final lista para su despliegue en un entorno de producción debido a las limitaciones mencionadas, el diseño e implementación de los métodos de RAG y la arquitectura de datos son sólidos, correctos y escalables. Por tanto, se considera que el proyecto ha alcanzado satisfactoriamente los objetivos planteados dentro de su alcance. Además, representa un ejemplo válido del potencial de la inteligencia artificial aplicada a este tipo de herramientas, sentando las bases para futuros desarrollos en entornos reales.

Desde una perspectiva personal, el desarrollo de este proyecto ha supuesto una significativa oportunidad de aprendizaje, permitiendo la adquisición de conocimientos fundamentales. A lo largo del proceso, se ha logrado una comprensión sólida de conceptos clave, como las fases de entrenamiento de este tipo de modelo, la arquitectura de los *transformers*, así como de los principios que rigen el funcionamiento de los modelos de lenguaje de gran escala. Asimismo, se ha alcanzado la capacidad de implementar una solución funcional basada en estas tecnologías, lo cual ha permitido consolidar y aplicar de forma práctica los conocimientos teóricos adquiridos.

6.3. Impacto del proyecto

El desarrollo de este proyecto ha supuesto una contribución significativa al proceso de digitalización del sector cultural. Tal y como se expuso en el capítulo 1, uno de los objetivos fundamentales de este trabajo era proponer una solución innovadora, basada en inteligencia artificial, que pudiera integrarse eficazmente en entornos culturales para mejorar la accesibilidad, la interacción y el valor percibido por el visitante.

La plataforma desarrollada no solo refleja un avance en la aplicación de tecnologías digitales dentro del ámbito cultural, sino que también introduce mejoras sustanciales en la experiencia de usuario. A través de una interfaz conversacional intuitiva y personalizable, se consigue transformar la forma en que los visitantes interactúan con los contenidos museísticos, generando una experiencia más enriquecedora, cercana y dinámica.

En este sentido, el impacto del proyecto se manifiesta en tres ejes fundamentales: la promoción de la cultura mediante nuevas tecnologías, el impulso de la digitalización del patrimonio, y la mejora en la calidad y personalización de la experiencia del

usuario en contextos museísticos.

Las decisiones adoptadas a lo largo del desarrollo del proyecto han estado orientadas a maximizar el impacto en el proceso de digitalización del entorno museístico. En particular, se ha prestado especial atención a la estructuración modular de las herramientas que conforman el sistema. Esta modularidad no solo facilita su mantenimiento y escalabilidad, sino que también permite su adaptación e integración progresiva en distintas fases del proceso de digitalización, desde la implementación de funcionalidades básicas hasta la incorporación de asistentes conversacionales avanzados.

Asimismo, se han priorizado decisiones técnicas orientadas a la eficiencia computacional y a la optimización del consumo de recursos. Estas medidas no solo permiten mejorar el rendimiento del sistema, sino que también inciden directamente en la reducción de costes asociados al uso de modelos de lenguaje de gran tamaño. Esta consideración resulta especialmente relevante en contextos de producción real, donde el equilibrio entre rendimiento, coste y escalabilidad es esencial para la viabilidad del sistema.

6.4. Trabajo futuro

De cara a desarrollos posteriores, se identifican las siguientes líneas de mejora y expansión del sistema propuesto.

- **Optimización de las técnicas avanzadas de RAG:** Las técnicas de RAG constituyen la base sobre la que se sustenta el funcionamiento del chatbot, y por tanto, representan un componente crítico del sistema. En este sentido, se plantea como línea prioritaria la mejora de dichas técnicas, centrándose en diversos aspectos: una mejor estructuración de los datos para facilitar una recuperación más precisa del contexto; el perfeccionamiento de los procesos previos a la recuperación de información, con el objetivo de interpretar con mayor precisión las preguntas formuladas por los usuarios; y la optimización de la inserción del contexto en la conversación, de forma que se minimice el consumo de *tokens*.
- **Ampliación y enriquecimiento de las fuentes de información:** Actualmente, la base de datos vectorial se nutre exclusivamente de información proporcionada por el propio museo y sin modificar. Si bien esta es suficiente para muchas obras, en otros casos resulta escasa o poco significativa para el visitante. Por ello, se plantea como línea futura la incorporación de nuevas fuentes de información que complementen y enriquezcan el conocimiento disponible. Esta ampliación deberá considerar no solo el contenido informativo, sino también la forma en que los visitantes suelen estructurar sus consultas, con el fin de crear datos más adaptados a las necesidades reales de los usuarios.
- **Aplicación de técnicas de ajuste fino sobre los modelos:** Tal como se ha mencionado previamente, esta opción fue descartada debido a las limitaciones de capacidad computacional y a la falta de datos adecuados. No obstante, llevar a cabo un proceso de ajuste fino permitiría adaptar el modelo a casos de uso específicos, mejorar la precisión de sus respuestas, y reducir la dependencia de proveedores externos, como Groq, lo cual repercutiría positivamente en la autonomía y eficiencia del sistema.
- **Presentación de resultados a entidades culturales, como el Museo Nacional Thyssen-Bornemisza:** A pesar de que la versión actual del sistema no se

Resultados y conclusiones

considera aún apta para su despliegue en producción —dado que no ha sido evaluada en un amplio rango de escenarios y su rendimiento depende en parte de servicios gratuitos limitados—, esta constituye una prueba de concepto. Por ello, se propone como objetivo a corto o medio plazo la presentación del proyecto al Museo Nacional Thyssen-Bornemisza, u otras instituciones culturales, con el fin de explorar posibles colaboraciones y valorar su aplicación práctica en entornos reales.

- **Despliegue en producción:** A largo plazo, y bajo una perspectiva optimista, se contempla la posibilidad de llevar a cabo el despliegue del sistema en un entorno de producción, permitiendo su uso por parte del público visitante del museo. Este objetivo requeriría no solo una validación exhaustiva en condiciones reales, sino también la optimización de los recursos técnicos y económicos necesarios para su mantenimiento.

Bibliografía

- [1] OpenAI. *Introducing ChatGPT*. 2022. URL: <https://openai.com/index/chatgpt/>.
- [2] Revista de Arte. *El Museo del Prado utiliza inteligencia artificial en Internet*. Oct. de 2019. URL: <https://www.revistadearte.com/2019/10/08/el-museo-del-prado-utiliza-inteligencia-artificial-en-internet>.
- [3] Óscar Granados. *Así se logra que un museo tenga éxito*. Nov. de 2024. URL: <https://elpais.com/economia/negocios/2024-11-03/asi-se-logra-que-un-museo-tenga-exito.html>.
- [4] Museo Nacional Thyssen-Bornemisza. *Museo Nacional Thyssen-Bornemisza*. 2025. URL: <https://www.museothyssen.org/>.
- [5] Pedro Larrañaga et al. *Industrial Applications of Machine Learning*. CRC Press, 2019. DOI: 10.1201/9781351128384.
- [6] Ashish Vaswani et al. «Attention Is All You Need». En: *CoRR abs/1706.03762 (2017)*. arXiv: 1706.03762. URL: <http://arxiv.org/abs/1706.03762>.
- [7] IBM. *¿Qué son las redes neuronales?* s.f. URL: <https://www.ibm.com/es-es/topics/neural-networks>.
- [8] Wikipedia. *Ajuste fino (aprendizaje profundo) — Wikipedia, La enciclopedia libre*. 2025. URL: [https://es.wikipedia.org/w/index.php?title=Ajuste_fino_\(aprendizaje_profundo\)&oldid=164550731](https://es.wikipedia.org/w/index.php?title=Ajuste_fino_(aprendizaje_profundo)&oldid=164550731).
- [9] Wikipedia contributors. *Reinforcement learning from human feedback — Wikipedia, The Free Encyclopedia*. 2025. URL: https://en.wikipedia.org/w/index.php?title=Reinforcement_learning_from_human_feedback&oldid=1284965638.
- [10] Neil Houlsby et al. *Parameter-Efficient Transfer Learning for NLP*. 2019. arXiv: 1902.00751 [cs.LG]. URL: <https://arxiv.org/abs/1902.00751>.
- [11] Wikipedia. *Word embedding — Wikipedia, La enciclopedia libre*. 2024. URL: https://es.wikipedia.org/w/index.php?title=Word_embedding&oldid=159890633.
- [12] Bang Liu et al. *Advances and Challenges in Foundation Agents: From Brain-Inspired Intelligence to Evolutionary, Collaborative, and Safe Systems*. 2025. arXiv: 2504.01990 [cs.AI]. URL: <https://arxiv.org/abs/2504.01990>.
- [13] Zixiang Chen et al. *Towards Understanding Mixture of Experts in Deep Learning*. 2022. arXiv: 2208.02813 [cs.LG]. URL: <https://arxiv.org/abs/2208.02813>.
- [14] Dan Hendrycks et al. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: <https://arxiv.org/abs/2009.03300>.
- [15] Xiang Yue et al. *MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI*. 2024. arXiv: 2311.16502 [cs.CL]. URL: <https://arxiv.org/abs/2311.16502>.

- [16] OpenAI. *GPT-4.5 System Card*. Feb. de 2025. URL: <https://openai.com/index/gpt-4-5-system-card/>.
- [17] Reem Aleithan et al. *SWE-Bench+: Enhanced Coding Benchmark for LLMs*. 2024. arXiv: 2410.06992 [cs.SE]. URL: <https://arxiv.org/abs/2410.06992>.
- [18] Shanghaoran Quan et al. *CodeElo: Benchmarking Competition-level Code Generation of LLMs with Human-comparable Elo Ratings*. 2025. arXiv: 2501.01257 [cs.CL]. URL: <https://arxiv.org/abs/2501.01257>.
- [19] OpenAI. *OpenAI o3 and o4-mini System Card*. Consultado el 14 de mayo de 2025. Abr. de 2025. URL: <https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf>.
- [20] Meta AI. *Llama 4*. 2025. URL: <https://www.llama.com/models/llama-4/#models>.
- [21] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: <https://arxiv.org/abs/2501.12948>.
- [22] xAI. *Grok: AI Model by xAI*. 2023. URL: <https://x.ai/grok>.
- [23] Anthropic. *Claude 3.7 Sonnet*. 2025. URL: <https://www.anthropic.com/claude/sonnet>.
- [24] Xinyu Zhang et al. *Making a MIRACL: Multilingual Information Retrieval Across a Continuum of Languages*. 2022. arXiv: 2210.09984 [cs.IR]. URL: <https://arxiv.org/abs/2210.09984>.
- [25] Niklas Muennighoff et al. *MTEB: Massive Text Embedding Benchmark*. 2023. arXiv: 2210.07316 [cs.CL]. URL: <https://arxiv.org/abs/2210.07316>.
- [26] OpenAI. *New embedding models and API updates*. Accedido el 16 de mayo de 2025. Ene. de 2024. URL: <https://openai.com/index/new-embedding-models-and-api-updates/>.
- [27] Zach Nussbaum et al. *Nomic Embed: Training a Reproducible Long Context Text Embedder*. 2025. arXiv: 2402.01613 [cs.CL]. URL: <https://arxiv.org/abs/2402.01613>.
- [28] Xinyu Zhang et al. *Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval*. 2021. arXiv: 2108.08787 [cs.CL]. URL: <https://arxiv.org/abs/2108.08787>.
- [29] Liang Wang et al. *Multilingual E5 Text Embeddings: A Technical Report*. 2024. arXiv: 2402.05672 [cs.CL]. URL: <https://arxiv.org/abs/2402.05672>.
- [30] Georgios Trichopoulos et al. «Large Language Models as Recommendation Systems in Museums». En: *Electronics* 12.18 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12183829. URL: <https://www.mdpi.com/2079-9292/12/18/3829>.
- [31] ArtDaily. *The Centre Pompidou launches its first Chatbot*. 2020. URL: <https://artdaily.com/news/142572/The-Centre-Pompidou-launches-its-first-Chatbot-#.Yiju2XrMKU1>.
- [32] Centre Pompidou. *Chatbot du Centre Pompidou*. 2025. URL: <https://www.centrepompidou.fr/fr/chatbot>.
- [33] Askmona. *Askmona Chatbot*. 2025. URL: <https://www.askmona.fr/en/chatbot/>.
- [34] Iva Vasic. «Large Language Models for Museums: Benefit or Threat?» En: jun. de 2024, págs. 203-206. ISBN: 9788899586423.
- [35] Zhan Wang et al. «VirtuWander: Enhancing Multi-modal Interaction for Virtual Tour Guidance through Large Language Models». En: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300.

- DOI: 10.1145/3613904.3642235. URL: <https://doi.org/10.1145/3613904.3642235>.
- [36] Khanh-Tung Tran et al. *Multi-Agent Collaboration Mechanisms: A Survey of LLMs*. 2025. arXiv: 2501.06322 [cs.AI]. URL: <https://arxiv.org/abs/2501.06322>.
- [37] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. arXiv: 2312.10997 [cs.CL]. URL: <https://arxiv.org/abs/2312.10997>.
- [38] Yuntong Hu et al. *GRAG: Graph Retrieval-Augmented Generation*. 2024. arXiv: 2405.16506 [cs.LG]. URL: <https://arxiv.org/abs/2405.16506>.
- [39] Wikipedia. *Python* — *Wikipedia, La enciclopedia libre*. 2025. URL: <https://es.wikipedia.org/w/index.php?title=Python&oldid=166821272>.
- [40] Wikipedia. *Web scraping* — *Wikipedia, La enciclopedia libre*. 2025. URL: https://es.wikipedia.org/w/index.php?title=Web_scraping&oldid=165248270.
- [41] Wikipedia. *Beautiful Soup* — *Wikipedia, La enciclopedia libre*. 2024. URL: https://es.wikipedia.org/w/index.php?title=Beautiful_Soup&oldid=157327514.
- [42] Wikipedia contributors. *Llama.cpp* — *Wikipedia, The Free Encyclopedia*. 2025. URL: <https://en.wikipedia.org/w/index.php?title=Llama.cpp&oldid=1282783088>.
- [43] Wikipedia. *Kit de desarrollo de software* — *Wikipedia, La enciclopedia libre*. [Internet; descargado 16-septiembre-2023]. 2023. URL: https://es.wikipedia.org/w/index.php?title=Kit_de_desarrollo_de_software&oldid=153776134.
- [44] LlamaIndex. *LlamaIndex* — *Build Knowledge Assistants over your Enterprise Data*. 2025. URL: <https://www.llamaindex.ai/>.
- [45] Wikipedia contributors. *MongoDB* — *Wikipedia, The Free Encyclopedia*. 2025. URL: <https://en.wikipedia.org/w/index.php?title=MongoDB&oldid=1285333989>.
- [46] Pinecone. *Rerankers and Two-Stage Retrieval*. 2024. URL: <https://www.pinecone.io/learn/series/rag/rerankers/>.
- [47] Zach Nussbaum et al. *Nomic Embed: Training a Reproducible Long Context Text Embedder*. Inf. téc. Nomic AI, 2024. URL: https://static.nomic.ai/reports/2024_Nomic_Embed_Text_Technical_Report.pdf.
- [48] Hao Yu et al. «Evaluation of Retrieval-Augmented Generation: A Survey». En: *Big Data*. Springer Nature Singapore, 2025, págs. 102-120. ISBN: 9789819610242. DOI: 10.1007/978-981-96-1024-2_8. URL: http://dx.doi.org/10.1007/978-981-96-1024-2_8.
- [49] Christian Grech. *Evaluating RAG Performance: A Comprehensive Guide*. 2024. URL: <https://christiangrech.medium.com/evaluating-rag-performance-a-comprehensive-guide-b1d8f903b7ad>.
- [50] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: <https://arxiv.org/abs/2303.08774>.
- [51] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: <https://arxiv.org/abs/2407.21783>.

Apéndice A

Apéndice de información incluida en Museion

A.1. Autores incluidos

- Alberto Durero
- Berthe Morisot
- Canaletto (Giovanni Antonio Canal)
- Caravaggio (Michelangelo Merisi)
- Claude Monet
- Duccio di Buoninsegna
- Edgar Degas
- Edward Hopper
- El Greco (Doménikos Theotokópoulos)
- Édouard Manet
- Ernst Ludwig Kirchner
- Francisco de Goya
- Georgia O'Keeffe
- Harmensz van Rijn Rembrandt
- Jan van Eyck
- Joan Miró
- Paul Gauguin
- Piet Mondrian
- Salvador Dalí
- Velázquez (Diego Rodríguez de Silva y Velázquez)
- Vincent van Gogh

A.2. Obras maestras incluidas

- *Cristo y la samaritana*, Duccio di Buoninsegna
- *Retrato de Giovanna degli Albizzi Tornabuoni*, Domenico Ghirlandaio
- *Jesús entre los doctores*, Alberto Durero
- *Joven caballero en un paisaje*, Vittore Carpaccio
- *La ninfa de la fuente*, Lucas Cranach el Viejo
- *El Gran Canal desde San Vio, Venecia*, Canaletto
- *Retrato de Enrique VIII de Inglaterra*, Hans Holbein el Joven
- *La Anunciación*, El Greco (Doménikos Theotokópoulos)
- *Santa Catalina de Alejandría*, Caravaggio (Michelangelo Merisi)
- *Venus y Cupido*, Peter Paul Rubens
- *Autorretrato con gorra y dos cadenas*, Rembrandt (Harmensz. van Rijn)
- *Grupo familiar en un paisaje*, Frans Hals
- *Fachada occidental de la iglesia de Santa María en Utrecht*, Pieter Jansz Saenredam
- *El columpio*, Jean-Honoré Fragonard
- *Bailarina basculando (La bailarina verde)*, Edgar Degas
- *La amazona de frente*, Édouard Manet
- *Los Vessenots en Auvers*, Vincent van Gogh
- *Retrato de Millicent, duquesa de Sutherland*, John Singer Sargent
- *Hombre sentado*, Paul Cézanne
- *El puente de Waterloo*, André Derain
- *Fränzi en una silla tallada*, Ernst Ludwig Kirchner
- *El sueño*, Franz Marc
- *Metrópolis*, George Grosz
- *Casa giratoria*, Paul Klee
- *Arlequín con espejo*, Pablo Picasso
- *Retrato del Dr. Haustein*, Christian Schad
- *Habitación de hotel*, Edward Hopper
- *Quappi con suéter rosa*, Max Beckmann
- *Sueño causado por el vuelo de una abeja alrededor de una granada un segundo antes de despertar*, Salvador Dalí
- *Mujer en el baño*, Roy Lichtenstein
- *Retrato de George Dyer en un espejo*, Francis Bacon
- *Hombre rojo con bigote*, Willem de Kooning

A.3. Recorridos temáticos incluidos

1. **Amor diverso**, compuesto por:

- *Retrato de un joven como San Sebastián*, Bronzino (Agnolo di Cosimo di Mariano)
- *Hércules en la corte de Onfalia*, Hans Cranach
- *Santa Catalina de Alejandría*, Caravaggio (Michelangelo Merisi)
- *San Sebastián*, Gian Lorenzo Bernini
- *David con la cabeza de Goliat y dos soldados*, Valentin de Boulogne
- *Muchacho con turbante y un ramillete de flores*, Michiel Sweerts
- *La muerte de Jacinto*, Giambattista Tiepolo
- *Amazona de frente*, Édouard Manet
- *Retrato de Millicent, duquesa de Sutherland*, John Singer Sargent
- *El nacimiento de Venus (La Aurora)*, Auguste Rodin
- *Desnudo*, Marc Chagall
- *Love, Love, Love. Homenaje a Gertrude Stein*, Charles Demuth
- *En memoria de Cecchino Bracci*, David Hockney
- *Retrato de George Dyer en un espejo*, Francis Bacon
- *Express*, Robert Rauschenberg
- *El griego de Esmirna (Nikos)*, Ronald B. Kitaj

2. **La cultura del vino**, compuesto por:

- *La Virgen y el Niño con un racimo de uvas*, Lucas Cranach el Viejo
- *La Sagrada Familia con un ángel*, Jacob Jordaens y taller
- *La Sagrada Familia con ángeles y santos*, Joachim Antonisz Wtewael
- *Baco y Ariadna*, Sebastiano Ricci
- *Bodegón con frutas*, Louise Moillon
- *El vino en la fiesta de San Martín*, Pieter Brueghel el Joven
- *La vendimia*, Francisco de Goya
- *El brindis*, Jan Steen
- *Bodegón con copa de vino*, Willem Claesz. Heda
- *Bodegón con frutas y copa de vino*, Jan Davidsz de Heem
- *Bodegón con uvas*, Juan de Espinosa
- *La cata de vinos*, David Teniers el Joven
- *Interior con bebedores*, Adriaen van Ostade
- *El almuerzo*, Jean-Baptiste-Siméon Chardin

- *La merienda*, Luis Meléndez
- *La bodega*, Francisco de Zurbarán
- *Bodegón con frutas y vino*, Bartolomé Esteban Murillo
- *La taberna*, Diego Velázquez
- *El bebedor alegre*, Frans Hals

3. **La moda**, compuesto por:

- *La Adoración del Niño*, Jacques Daret
- *Retrato de Giovanna degli Albizzi Tornabuoni*, Domenico Ghirlandaio (Domenico Bigordi)
- *Retrato de Enrique VIII de Inglaterra*, Hans Holbein el Joven
- *Retrato de una dama*, Hans Baldung Grien
- *Santa Casilda*, Zurbarán
- *Retrato de una joven dama con rosario*, Peter Paul Rubens
- *Retrato de David Lyon*, Thomas Lawrence
- *Joven con vestido japonés. El kimono*, William Merritt Chase
- *Amazona de frente*, Édouard Manet
- *En la sombrerería*, Edgar Degas
- *Desfile de modelos*, Lovis Corinth
- *Quappi con suéter rosa*, Max Beckmann
- *El fumador (Frank Haviland)*, Juan Gris
- *Vestidos simultáneos. (Tres mujeres, formas, colores)*, Sonia Delaunay
- *Nueva York, 3 (inacabado)*, Piet Mondrian

4. **Un día en Nueva York**, compuesto por:

- *Una clientela dura*, John George Brown
- *Niños en la playa*, Samuel S. Carr
- *La Quinta Avenida en Washington Square, Nueva York*, Childe Hassam
- *En el parque (Un camino)*, William Merritt Chase
- *Surtidor en Madison Square*, John Sloan
- *Metrópolis*, George Grosz
- *Battery Park*, Reginald Marsh
- *Habitación de hotel*, Edward Hopper
- *Estación terminal "Grand Central"*, Max Weber
- *En el puerto*, Albert Gleizes
- *Nueva York, 3 (inacabado)*, Piet Mondrian
- *Sin título (Verde sobre morado)*, Mark Rothko

- *Nedick's*, Richard Estes
- *People's Flowers*, Richard Estes
- *Cabinas telefónicas*, Richard Estes

5. **Gastronomía**, compuesto por:

- *Vista de la Carrera de San Jerónimo y el Paseo del Prado con cortejo de carrozas*, Jan van Kessel III (atribuido a)
- *La Última Cena*, Maestro de la Virgo inter Virgines (seguidor del)
- *Retrato de una joven*, Paris Bordone
- *Retrato de Matthäus Schwarz*, Christoph Amberger
- *Adán y Eva*, Jan Gossaert (llamado Mabuse)
- *La Última Cena*, Anónimo veneciano
- *Paisaje idílico con la huida a Egipto*, Claudio de Lorena
- *Esau vendiendo su primogenitura*, Hendrick ter Brugghen
- *Vertumno y Pomona*, Caesar van Everdingen
- *El antiguo mercado del pescado en el Dam, Amsterdam*, Emanuel de Witte
- *La cocinera*, Gabriel Metsu
- *Comiendo ostras*, Jacob Lucasz. Ochtervelt
- *Bodegón con pastel de frutas y diversos objetos*, Willem Claesz Heda
- *Bodegón con loza y dulces*, Juan van der Hamen y León
- *Bodegón con gato y raya*, Jean Baptiste Siméon Chardin
- *Manifestación patriótica*, Giacomo Balla
- *Nedick's*, Richard Estes

6. **Joyas**, compuesto por:

- *Retrato de Enrique VIII de Inglaterra*, Hans Holbein el Joven
- *Retrato de Giovanna degli Albizzi Tornabuoni*, Domenico Ghirlandaio (Domenico Bigordi)
- *Retrato de un hombre con una sortija*, Francesco del Cossa
- *La Virgen y el Niño con las santas Margarita y Catalina*, Michael Pacher (seguidor de)
- *Retrato de una joven dama con rosario*, Peter Paul Rubens
- *Venus y Cupido*, Peter Paul Rubens
- *Retrato de grupo con sir Elijah y lady Impey*, Johan Zoffany
- *Retrato de Millicent, duquesa de Sutherland*, John Singer Sargent
- *Quappi con suéter rosa*, Max Beckmann
- *Fränzi ante una silla tallada*, Ernst Ludwig Kirchner

7. **Pioneras. Mujeres artistas de la vanguardia rusa**, compuesto por:

- *Pesca (pescadores)*, Natalia Goncharova
- *El bosque*, Natalia Goncharova
- *Composición con rectángulo azul*, Natalia Goncharova
- *Naturaleza muerta*, Alexandra Exter
- *Hombre en la calle (Análisis de volúmenes)*, Olga Rózanova
- *Cubismo*, Nadeshda Udaltsova
- *Arquitectura pictórica (Bodegón: Instrumentos)*, Liubov Popova
- *Arquitectura pictórica*, Liubov Popova
- *Jugadores de billar*, Varvara Fedorovna Stepanova
- *Contrastes simultáneos*, Sonia Delaunay
- *Vestidos simultáneos. (Tres mujeres, formas, colores)*, Sonia Delaunay

8. **Pintura flamenca, neerlandesa y holandesa en la colección**, compuesto por:

- *Díptico de la Anunciación*, Jan van Eyck
- *Retrato de un hombre robusto (¿Robert de Masmines?)*, Robert Campin
- *Retrato de un hombre joven orante (anverso)*, Hans Memling
- *Florero (reverso)*, Hans Memling
- *Adán y Eva*, Jan Gossaert (llamado Mabuse)
- *Paisaje con el descanso en la huida a Egipto*, Joachim Patinir
- *Retrato de Jacques Le Roy*, Anton van Dyck
- *Venus y Cupido*, Peter Paul Rubens
- *Autorretrato con gorra y dos cadenas*, Harmensz van Rijn Rembrandt
- *El tamborilero desobediente*, Nicolaes Maes
- *Vista panorámica con ciudad al fondo*, Philips Koninck
- *Bodegón con cuenco chino, copa nautilo y otros objetos*, Willem Kalf
- *Les Vessenots en Auwers*, Vincent van Gogh
- *Nueva York, 3 (inacabado)*, Piet Mondrian
- *La llave de los campos (La Clef des champs)*, René Magritte

9. **Ingenios**, compuesto por:

- *El evangelista San Juan*, Gabriel Mälesskircher
- *Joven caballero en un paisaje*, Vittore Carpaccio
- *Jesús entre los doctores*, Alberto Durero
- *Retrato de una dama hilando*, Maerten van Heemskerck
- *Santa Catalina de Alejandría*, Caravaggio (Michelangelo Merisi)

- *El Gran Canal desde San Vío, Venecia*, Canaletto (Giovanni Antonio Canal)
- *Escena de invierno con patinadores y trineos ante una ciudad*, Salomon Jacobsz van Ruysdael
- *Paisaje invernal con pueblo, patinadores en un río helado y cazadores en primer plano*, Jacob Grimmer
- *Molino de agua en Gennep*, Vincent van Gogh
- *Las cuatro estaciones: invierno*, Pierre-Antoine Quillard
- *Molino cerca de Delft*, Johan Barthold Jongkind
- *El Sena en el puente de Sèvres*, Stanislas Lépine
- *Fábrica a la luz de la luna*, Maximilien Luce
- *Canal du Midi*, Henri Matisse
- *Proun 1 C*, El Lissitzky
- *Gran pintura del ferrocarril*, László Moholy-Nagy
- *Cabinas telefónicas*, Richard Estes

10. **Un paseo entre las flores**, compuesto por:

- *La Virgen con el Niño entronizada*, Rogier van der Weyden
- *Florero (reverso)*, Hans Memling
- *Retrato de una infanta (¿Catalina de Aragón?)*, Juan de Flandes
- *Retrato de una joven*, Paris Bordone
- *Joven caballero en un paisaje*, Vittore Carpaccio
- *Santa Casilda*, Zurbarán
- *La muerte de Jacinto*, Giambattista Tiepolo
- *El Nieuwezijds Voorburgwal con el mercado de flores de Amsterdam*, Gerrit Adriaensz Berckheyde
- *Vaso chino con flores, conchas e insectos*, Ambrosius Bosschaert I
- *Crisantemos en un florero*, Henri Fantin-Latour
- *Las flores amarillas*, Henri Matisse
- *Lirio blanco n° 7*, Georgia O'Keeffe
- *People's Flowers*, Richard Estes

11. **El viaje**, compuesto por:

- *La Adoración de los Reyes*, Luca di Tommè
- *Los argonautas abandonan la Cólquida*, Ercole Ferrarese
- *Retrato del dux Francesco Venier*, Tiziano (Tiziano Vecellio)
- *Paisaje idílico con la huida a Egipto*, Claudio de Lorena
- *El Gran Canal desde San Vío, Venecia*, Canaletto (Giovanni Antonio Canal)

A.3. Recorridos temáticos incluidos

- *Vista idealizada con ruinas romanas, esculturas y un puerto*, Bartholomeus Breenbergh
- *La Cena de Emaús*, Matthias Stom
- *Rincón de una biblioteca*, Jan van der Heyden
- *Paisaje montañoso con un castillo*, Roelandt Savery
- *Vista de las ruinas de Olinda, Brasil*, Frans Jansz Post
- *La flota holandesa en de Goeree*, Willem van de Velde II
- *Bodegón con cuenco chino, copa nautilo y otros objetos*, Willem Kalf
- *El patio de la aduana*, Nicolas-Bernard Lépicie
- *Retrato de grupo con sir Elijah y lady Impey*, Johan Zoffany
- *Las cataratas de San Antonio, Alto Mississippi*, Henry Lewis
- *Joven con vestido japonés. El kimono*, William Merritt Chase
- *Jinete árabe*, Eugène Delacroix
- *Los descargadores en Arlés*, Vincent van Gogh
- *Habitación de hotel*, Edward Hopper
- *Luces V. El pabellón del muelle*, Michael Andrews
- *El griego de Esmirna (Nikos)*, Ronald B. Kitaj
- *Sillón nº 2*, Domenico Gnoli

12. **Nubla**, compuesto por:

- *Retrato de una dama*, Hans Baldung Grien
- *Doble retrato de Hilde II*, Karl Hubbuch
- *Expulsión. Luna y luz de fuego*, Thomas Cole
- *Mañana de Pascua*, Caspar David Friedrich
- *Habitación de hotel*, Edward Hopper
- *La casa de la esquina (Villa Kochmann en Dresde)*, Ludwig Meidner
- *Metrópolis*, George Grosz
- *Burbuja de jabón azul*, Joseph Cornell
- *Árbol solitario y árboles conyugales*, Max Ernst
- *El proscrito deslumbrante. Del ciclo: El proscrito deslumbrante*, Matta

13. **Las ciudades invisibles**, compuesto por:

- *La Crucifixión*, Gerard David
- *El Gran Canal desde San Vio, Venecia*, Canaletto (Giovanni Antonio Canal)
- *Interior del templo de Diana en Nîmes*, Hubert Robert
- *Rue Saint-Honoré por la tarde. Efecto de lluvia*, Camille Pissarro
- *Casas junto al río. La ciudad vieja*, Egon Schiele

- *La dama de malva*, Lyonel Feininger
- *Escena callejera (Kurfürstendamm)*, George Grosz
- *Parque de atracciones*, Ben Shahn
- *Nueva York, 3 (inacabado)*, Piet Mondrian
- *Composición suprematista*, Ilyá Chashnik
- *La ciudad*, Wols
- *Cabinas telefónicas*, Richard Estes

14. **Placeres de la vida**, compuesto por:

- *Los jugadores de cartas*, Lucas Hugensz van Leyden
- *La carta amorosa*, François Clouet
- *Retrato del Conde Fulvio Grati*, Giuseppe Maria Crespi
- *Las cosquillas*, Pietro Longhi
- *Fumadores en un interior*, David Teniers II
- *La sala del concejo del Ayuntamiento de Amsterdam*, Pieter Hendricksz de Hooch
- *Paisaje invernal con figuras en el hielo*, Jan Josephsz van Goyen
- *Retrato de grupo con sir Elijah y lady Impey*, Johan Zoffany
- *Pierrot contento*, Jean Antoine Watteau
- *El columpio*, Jean-Honoré Fragonard

15. **Eva al desnudo**, compuesto por:

- *El nacimiento de Venus (La Aurora)*, Auguste Rodin
- *La Crucifixión*, Vitale da Bologna
- *La ninfa de la fuente*, Lucas Cranach el Viejo
- *Adán y Eva*, Hans Baldung Grien
- *La muerte de Jacinto*, Giambattista Tiepolo
- *Venus y Marte*, Carlo Saraceni
- *El duque de Orleans mostrando a su amante*, Eugène Delacroix
- *Dos desnudos femeninos en un paisaje*, Otto Mueller
- *Las estampas*, Henri Manguin
- *Desnudo azul*, Mijaíl Lariónov
- *Sueño causado por el vuelo de una abeja alrededor de una granada un segundo antes del despertar*, Salvador Dalí
- *Express*, Robert Rauschenberg
- *Desnudo n^o 1*, Tom Wesselmann

16. **Un paseo por las artes escénicas**, compuesto por:

- *Pierrot contento*, Jean Antoine Watteau
- *Retrato de Ann Brown en el papel de Miranda*, Johan Zoffany
- *Bailarina basculando (Bailarina verde)*, Edgar Degas
- *Bailarinas en rosa*, Jean-Louis Forain
- *Yvette Guilbert*, Henri de Toulouse-Lautrec
- *Teatro de máscaras*, James Ensor
- *Jardin d'amour*, James Ensor
- *Varieté*, Georg Tappert
- *Circo*, August Macke
- *Manifestación patriótica*, Giacomo Balla
- *La casa gris*, Marc Chagall
- *Arlequín con espejo*, Pablo Picasso
- *Formación. Tripartición*, Oskar Schlemmer
- *El siglo XX*, Sándor Bortnyik
- *Habitación de hotel*, Edward Hopper
- *Smoko. El volcán humano*, Reginald Marsh
- *La llave de los campos (La Clef des champs)*, René Magritte
- *Chico con chistera*, Walt Kuhn
- *Luna sobre Alabama*, Richard Lindner
- *Caffè Greco*, Renato Guttuso
- *Retrato de hombre (Barón H. H. Thyssen-Bornemisza)*, Lucian Freud

17. **Una habitación propia**, compuesto por:

- *El nacimiento de la Virgen*, Jan de Beer
- *Retrato de una dama hilando*, Maerten van Heemskerck
- *Las cosquillas*, Pietro Longhi
- *Rincón de una biblioteca*, Jan Jansz van der Heyden
- *Retrato de un hombre leyendo un documento*, Gerard ter Borch
- *Interior con una mujer cosiendo y un niño*, Pieter Hendricksz de Hooch
- *El tamborilero desobediente*, Nicolaes Maes
- *La toilette*, François Boucher
- *Objetos para un rato de ocio*, William Michael Harnett
- *El duque de Orleans mostrando a su amante*, Eugène Delacroix
- *El espejo psiqué*, Berthe Morisot
- *El lector*, Ferdinand Hodler

- *Las estampas*, Henri Manguin
 - *Calle de Nueva York con luna*, Georgia O'Keeffe
 - *Habitación de hotel*, Edward Hopper
18. **Sostenibilidad. Algunos retos sociales en las obras de la colección**, compuesto por:
- *Paisaje idílico con la huida a Egipto*, Claudio de Lorena
 - *El Gran Canal desde San Vito, Venecia*, Canaletto (Giovanni Antonio Canal)
 - *Paisaje de invierno*, Jacob Isaacksz van Ruisdael y Colaboradores
 - *El lago George*, John Frederick Kensett
 - *Mañana de Pascua*, Caspar David Friedrich
 - *Los descargadores en Arlés*, Vincent van Gogh
 - *El puente de Waterloo*, André Derain
 - *Contrastes simultáneos*, Sonia Delaunay
 - *El bosque*, Natalia Goncharova
 - *Merzbild Kijkduin*, Kurt Schwitters
 - *Ritmos de la tierra*, Mark Tobey
 - *Domingo después del sermón*, Romare Bearden
19. **Las mujeres y los libros. Una reflexión a través de la pintura occidental**, compuesto por:
- *Díptico de la Anunciación*, Jan van Eyck
 - *La Anunciación*, Benedetto Bonfigli
 - *La Anunciación*, Gentile Bellini
 - *Retrato de Giovanna degli Albizzi Tornabuoni*, Domenico Ghirlandaio (Domenico Bigordi)
 - *La Virgen de la Anunciación / El ángel de la Anunciación (anverso)*, Bernhard Strigel
 - *La Anunciación*, Jan de Beer
 - *Retrato de una dama*, Pier Francesco Foschi
 - *Retrato de una mujer con un perro*, Veronés (Paolo Caliari)
 - *La Anunciación*, El Greco (Doménikos Theotokópoulos)
 - *Dama de cincuenta y seis años*, Cornelis Ketel
 - *Retrato de Helena de Kay*, Winslow Homer
 - *Joven con vestido japonés. El kimono*, William Merritt Chase
 - *Tarde de otoño*, Henri Le Sidaner
 - *Las estampas*, Henri Manguin
 - *Simone*, Georges D'Espagnat

- *Habitación de hotel*, Edward Hopper

20. **Camino del Agua. Arte y bienestar**, compuesto por:

- *Los argonautas abandonan la Cólquida*, Ercole de' Roberti
- *La ninfa de la fuente*, Lucas Cranach el Viejo
- *Paisaje con el descanso en la huida a Egipto*, Joachim Patinir
- *Paisaje idílico con la huida a Egipto*, Claudio de Lorena
- *Neptuno y Anftrite*, Sebastiano Ricci
- *Mar tormentoso con barcos de vela*, Jacob Isaacksz van Ruisdael
- *Mar tempestuoso*, Claude-Joseph Vernet
- *Escena en el jardín de un serrallo*, Giovanni Antonio Guardi
- *La pasarela*, Hubert Robert
- *El lago George*, John Frederick Kensett
- *El arroyo Brème*, Gustave Courbet
- *El baño de Diana (La Fuente)*, Jean-Baptiste-Camille Corot
- *El deshielo en Vétheuil*, Claude Monet
- *En el puerto*, Albert Gleizes
- *Flor-concha*, Max Ernst
- *Mujer en el baño*, Roy Lichtenstein