



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



## TFG del Grado en Ingeniería Informática

Eco City Tours: Aplicación móvil para la  
generación de rutas turísticas sostenibles  
propuestas por modelos de lenguaje a gran  
escala



Presentado por Fernando Pisot Serrano  
en Universidad de Burgos — 5 de septiembre  
de 2024

Tutor: Carlos López Nozal







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Carlos López Nozal, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Fernando Pisot Serrano, con DNI 70873328R, ha realizado el Trabajo final de Grado en Ingeniería Informática.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 5 de septiembre de 2024

Vº. Bº. del Tutor:

D. Carlos López Nozal tutor





## Resumen

Eco City Tours es una aplicación móvil **desarrollada con Flutter** que propone al usuario rutas turísticas sostenibles. La aplicación se enfoca en la promoción de rutas no motorizadas, optimizadas para ciclistas y peatones, que conectan lugares de interés con el objetivo de fomentar la movilidad sostenible. Las rutas se generan con tecnologías usando *Sistemas de Información Geográfica (GIS)*. Los *Puntos de Interés (POI)* se enriquecen con información detallada sobre lugares turísticos mediante inteligencia artificial consultando en lenguaje natural a *Modelos de Lenguaje a Gran Escala (LLM)*.

Esta aplicación se alinea con el concepto de **Smart City**, promoviendo activamente los *Objetivos de Desarrollo Sostenible (ODS)*, con un enfoque particular en el ODS11 (Ciudades y Comunidades Sostenibles).

## Descriptores

Movilidad Sostenible, LLM, ODS, Smart City, Flutter.

## Abstract

Eco City Tours is a mobile application **developed with Flutter** that suggests tourist routes to users. The application focuses on promoting non-motorized routes, optimized for cyclists and pedestrians, connecting points of interest with the goal of fostering sustainable mobility. The routes are generated using technologies based on *Geographic Information Systems (SIG)*. The *Points of Interest (POI)* are enriched with detailed information about tourist spots through artificial intelligence, consulting *Large Language Models (LLM)* via natural language queries.

This application aligns with the concept of **Smart City**, actively promoting the *Sustainable Development Goals (SDG)*, with a particular focus on the SDG11 (Sustainable Cities and Communities).

## Keywords

Sustainable Mobility, LLM, SDGs, Smart City, Flutter.



---

# Índice general

---

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos del proyecto</b>	<b>3</b>
2.1. Objetivos funcionales . . . . .	3
2.2. Objetivos Técnicos . . . . .	4
2.3. Objetivos Personales . . . . .	4
<b>3. Conceptos teóricos</b>	<b>5</b>
3.1. Large Language Models (LLM) . . . . .	5
3.2. Retrieval-Augmented Generation (RAG) . . . . .	7
3.3. Agentes . . . . .	9
3.4. Listas de items . . . . .	10
3.5. Tablas . . . . .	10
<b>4. Técnicas y herramientas</b>	<b>13</b>
4.1. LangChain . . . . .	13
<b>5. Aspectos relevantes del desarrollo del proyecto</b>	<b>15</b>
5.1. Elección de agentes . . . . .	16
5.2. Elección de servicios Google sobre tecnología OSM . . . . .	16
5.3. Elección de servicios Geocoding MapBox sobre servicios Google . . . . .	17

<b>6. Trabajos relacionados</b>	<b>19</b>
<b>7. Conclusiones y Líneas de trabajo futuras</b>	<b>25</b>
<b>Bibliografía</b>	<b>27</b>

---

# Índice de figuras

---

3.1. Preparación de la información de un RAG mostrada en la herramienta Langflow . . . . .	8
3.2. RAG mostrado en la herramienta Langflow usando <a href="https://astra.datastax.com">https://astra.datastax.com</a> . . . . .	9
6.1. Chatbot de Wanderlog . . . . .	20

---

# Índice de tablas

---

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	11
6.1. Aplicaciones Similares . . . . .	22
6.2. Comparación de aplicaciones similares . . . . .	23

---

# 1. Introducción

---

El crecimiento de población en las ciudades [5] y el turismo como catalizador de la gentrificación supone el gran campo de batalla para gobiernos locales de los países occidentales que han visto como la falta de una legislación controlada del turismo supone un grave problema afectando múltiples niveles de la convivencia, economía y el medio ambiente. A pesar de los avances en la promoción de un nuevo modelo urbano, muchas urbes aún enfrentan desafíos significativos en la integración de prácticas sostenibles en la vida cotidiana de sus habitantes. La falta de información accesible y personalizada sobre rutas y actividades que promuevan la movilidad sostenible y el turismo responsable es un marco común que se debe desarrollar si se quiere evitar que el conflicto crezca sin fin. Esta brecha de información impide que tanto residentes como turistas adopten hábitos más sostenibles que beneficien a la comunidad local y al medio ambiente en un marco global.

Fomentar el Turismo Sostenible supone una gran oportunidad para intentar contrarrestar la deriva actual. Y es que el turismo es un motor fundamental de la economía a nivel global y por tanto tiene la capacidad de transformarse para ayudar a la sostenibilidad del planeta. Destaca en este marco de trabajo el siguiente objetivo ODS identificado como número 11 que se centra en hacer que las ciudades y los asentamientos humanos sean sostenibles. Según el informe de la *UNESCO*, [2], no solo es crucial por sí mismo este ODS, sino que actúa como un multiplicador, influyendo indirectamente en la consecución de otros ODS debido a su enfoque integral y transversal.

En este contexto, la aplicación móvil desarrollada Eco City Tours **aúna estos esfuerzos al proporcionar una herramienta práctica y accesible** para la promoción del ODS11 y la movilidad sostenible. La aplicación Eco City Tours producto de este TFG, desarrollada en Flutter, utiliza

*Modelos de Lenguaje a Gran Escala (LLM)* para generar rutas turísticas personalizadas que conecten *Puntos de Interés (POI)*. La aplicación se enfoca en las preferencias del usuario, ofreciendo rutas optimizadas para ciclistas y peatones promoviendo así la movilidad sostenible. Eco City Tours durante el proceso de obtención de los *Puntos de Interés (POI)* requiere al **LLM** que utilice prácticas sostenibles como la deslocalización del turismo, esta solución no solo enriquece la experiencia turística [4], sino que también promueve prácticas sostenibles, que en lugar de perjudicar a los visitantes, puede apoyar el sustento de las comunidades locales. De este modo, se logra un impacto positivo tanto a nivel local como en el medio ambiente a escala global.

---

## 2. Objetivos del proyecto

---

### 2.1. Objetivos funcionales

Estos objetivos se centran en las funcionalidades y características que debe tener la aplicación para satisfacer las necesidades y expectativas de los usuarios. A continuación se detallan los objetivos funcionales del proyecto:

- **Propuesta de rutas turísticas personalizadas:** La aplicación debe ser capaz de generar rutas turísticas personalizadas basadas en las preferencias del usuario utilizando *Modelos de Lenguaje a Gran Escala (LLM)* para ello a través de la interfaz, el usuario facilitará al modelo sus preferencias por ejemplo si prefiere hacer la ruta a pie o en bicicleta, etc.
- **Obtener los *Puntos de Interés (POI)*:** La aplicación debe identificar y conectar diversos puntos de interés, proporcionando información relevante sobre cada uno.
- **Visualización de rutas en mapa:** La aplicación debe mostrar las rutas sugeridas en un mapa utilizando herramientas de gestión de información geográfica.
- **Optimización para ciclistas y peatones:** La aplicación debe promover la movilidad sostenible sugiriendo rutas optimizadas para ciclistas y peatones.
- **Interfaz Intuitiva y Amigable:** El usuario debe interactuar con la aplicación de manera intuitiva, siendo fácil de usar por los usuarios las diferentes funcionalidades.

## 2.2. Objetivos técnicos

Los objetivos técnicos se refieren a los desafíos y metas técnicas que se deben abordar para desarrollar el software. Estos objetivos abarcan aspectos como la arquitectura del sistema, las tecnologías a utilizar y las metodologías de desarrollo. A continuación se detallan los objetivos técnicos del proyecto:

- **Implementación de *Modelos de Lenguaje a Gran Escala (LLM)* y *LangChain*:** Integrar modelos de lenguaje a gran escala (*LLM*) y el framework *LangChain* para la generación de rutas y procesamiento de información relevante y ser capaz de integrar dicho conocimiento para ser mostrada en la aplicación móvil así como en un prototipo que muestre de manera incremental la mejora obtenida por parte de los modelos usando diferentes técnicas a la hora de interactuar con ellos como puede ser el uso de técnicas *Retrieval Augmented Generation (RAG)*, agentes, etc.
- **Uso de Herramientas Open-Source:** se priorizará para el desarrollo de la aplicación programas, paquetes o librerías que sean de código abierto siempre que sea posible y no incurra en un detrimento de la calidad del producto final.

## 2.3. Objetivos Personales

- **Conocimiento avanzado en *LLM*:** dada la rápida evolución, los amplios campos en los que se puede utilizar, etc. obtener una base de conocimientos destacable en este área sería un objetivo que me permitiría expandir mi futuro académico y por tanto distinguir mi perfil profesional especializándome en este sector en fuerte expansión.
- **Creación de aplicación móvil profesional:** de igual manera poner en práctica lo aprendido en varios cursos de *Dart y Flutter* puede contribuir a que la aplicación de este proyecto sea parte de mi portfolio con aplicaciones que muestren mis habilidades a futuros empleadores.
- **Finalización del *TFG* y Grado:** tras no haber completado la Ingeniería Técnica Informática en su momento por no haber realizado el Proyecto Fin de Carrera, la realización de este *TFG* marca la culminación de mi formación como ingeniero y supone el cierre de una carga personal.



---

## 3. Conceptos teóricos

---

En este capítulo se llevará a cabo una descripción de los conceptos necesarios para comprender el funcionamiento de la aplicación desarrollada. Es necesario tener un conocimiento acerca de los modelos del lenguaje que proporcionan el origen de datos en los que se basan los *Puntos de Interés (POI)*.

### 3.1. Large Language Models (LLM)

Empezamos por el concepto más general para luego ir acercándonos a la parte más concreta del desarrollo de la aplicación. Los modelos de lenguaje a gran escala es un tipo de inteligencia artificial que ha sido entrenada para comprender *Procesamiento del Lenguaje Natural (NLP)* que es la manera en que se comunican las personas. Estas inteligencias artificiales son entrenadas entonces con ingentes cantidades de datos que los hacen capaces de comprender peticiones, responder a las mismas en los mismos términos de lenguaje generando una especie de comunicación entre el usuario y la máquina.

#### Uso de LLMs en la aplicación de este TFG

En este trabajo el uso de los modelos de gran escala han sido usados para obtener los *Puntos de Interés (POI)*, basado en un juego de conversaciones con la inteligencia artificial el usuario determina basandose en el conocimiento del modelo qué lugares debería visitar a la hora de hacer turismo sostenible. En la sección de prototipos de este trabajo se observa como se va construyendo una comunicación con diferentes modelos: desde una conversación básica con resultados mediocres o incluso alucinados, hasta

construcciones que tienen en cuenta estructuras de datos que serán construidas como respuesta del modelo al usuario. La aplicación se beneficia de todo ello y genera una respuesta acorde al código que se quiere obtener en la aplicación móvil.

## Técnicas usadas en los prototipos

### Zero-shot y Few-shot learning

**Zero-shot** se trata de una técnica en la que el usuario no facilita al modelo ningún ejemplo de cómo realizar una tarea. El **LLM** por tanto interpreta basado en el contexto y su propio entrenamiento lo que se ha requerido y responde acorde a estos datos. Esta técnica se usa cuando lo que se prioriza es la rapidez del modelo frente a la precisión de la salida aportada. Cuando se requiere un trabajo de aproximación mayor una técnica que siempre mejora la conversación con el modelo es la técnica **few-shot learning**: se facilita en el prompt al modelo unos ejemplos de lo que se quiere obtener. Para comprenderlo mejor veamos el siguiente ejemplo de prompt:

Clasifica los siguientes comentarios como Positivos,  
Negativos o Neutros:

1. "El producto llegó a tiempo y en perfectas condiciones."  
Clasificación: Positivo

2. "El artículo no cumplió con mis expectativas,  
estoy decepcionado."  
Clasificación: Negativo

3. "La atención al cliente fue aceptable, pero podría mejorar."  
Clasificación: Neutro

4. "El servicio fue excelente, muy recomendable."  
Clasificación:

Al facilitar tres ejemplos de lo que se quiere obtener, la salida obtenida mejora y es lo que se espera por parte del usuario. Expresar en lenguaje natural lo que se quiere obtener es a veces más difícil y se puede malinterpretar por parte del modelo que darle unos ejemplos para que sepa con precisión el contexto. Más información al respecto se pueden observar en el prototipado

del proyecto. Para terminar de ajustar la salida obtenida se usa la siguiente técnica:

### Tool calling o function calling

Cuando la información del modelo tiene que ser muy precisa se recurre a esta técnica. En el caso del trabajo la información tenía que ser basada en una estructura que desde la programación se pudiera procesar fácilmente. Un archivo cuya estructura fuese en forma de json era vital. Para ello se le pide al modelo qué tipo de salida se requiere y para que no hubiese dudas se le facilitan un par de ejemplos. Una vez establecida la forma de la salida, se procede con el prompt de entrada usando la técnica que se quiera o cumpliendo con las especificaciones del modelo en concreto que se esté usando. De esta manera también se realiza una separación de abstracción que facilita la modularidad del código: se puede cambiar de origen en los datos, es decir, elegir otro modelo **LLM**, pero la salida del mismo siempre debe cumplir con estos requisitos desde el punto de vista de la programación. Es el mismo caso de abstracción usada en otros lenguajes de programación donde existe un repositorio y una fuente de datos. El programa se nutre de uno dejando el otro para acceder a datos de manera más concreta, donde el cambio de uno deja inalterado el funcionamiento del programa.

## 3.2. Retrieval-Augmented Generation (RAG)

Generación Aumentada por Recuperación es una técnica usada en modelos de inteligencia artificial en la cual se obtiene información para nutrir a un modelo de gran escala que ya ha sido entrenado, de esta manera amplía su conocimiento y es capaz de generar una respuesta más precisa, actualizada y completa. El problema que subyace en los modelos tradicionales es que una vez alimentados con un conjunto de datos, sufren de un aislamiento del mundo que los rodea.

Para prevenir este problema se nutre de información que el usuario facilita siguiendo los siguientes pasos:

1. **Splitter/tokenización:** la información proporcionada se mide en tokens y cada modelo tiene una cantidad que puede usar como contexto, además del coste que algunos modelos pueden cobrar al usuario por token, es por ello que transformar una cadena de texto inicial

que ocupa más espacio del estrictamente necesario en una cadena separada en pequeños trozos de información que además usa ciertos tokens especiales para mayor comprensión es una tarea previa a la recuperación de información.

2. **Embeddings:** consiste en transformar la información facilitada y representarla en vectores de  $n$  dimensiones. Para ellos se usa comúnmente otro modelo entrenado para transformar la información en vectores.

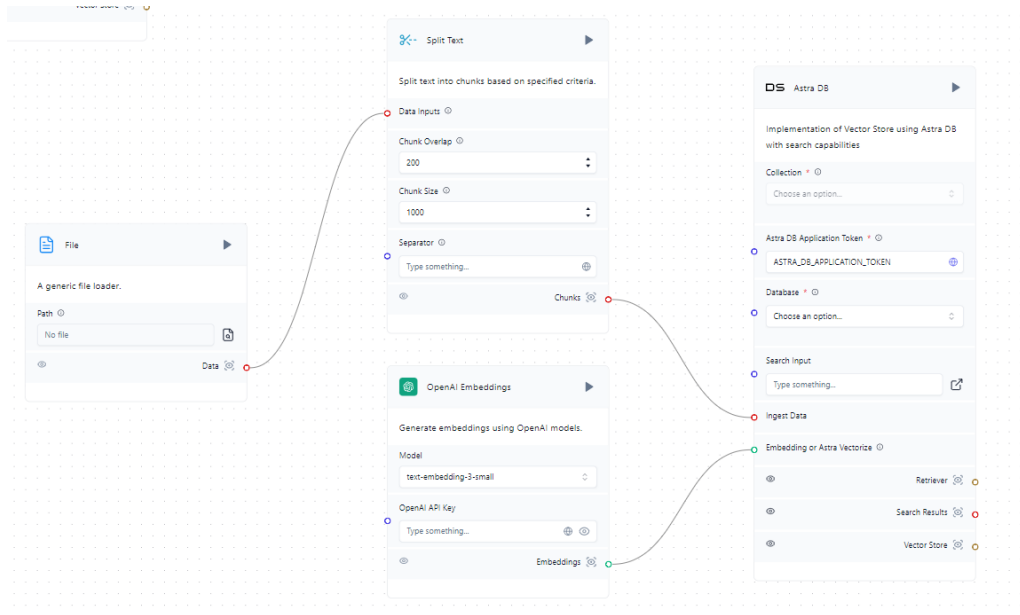


Figura 3.1: Preparación de la información de un RAG mostrada en la herramienta Langflow

3. **RAG:** con la información ampliada ya vectorizada en una base de datos, el usuario genera una entrada o prompt al modelo, el **LLM** entonces selecciona la información más afín de los datos aportados para generar así un prompt ampliado o mejorado que será pasado al modelo para un procesamiento de información habitual, consiguiendo así un mejor resultado.

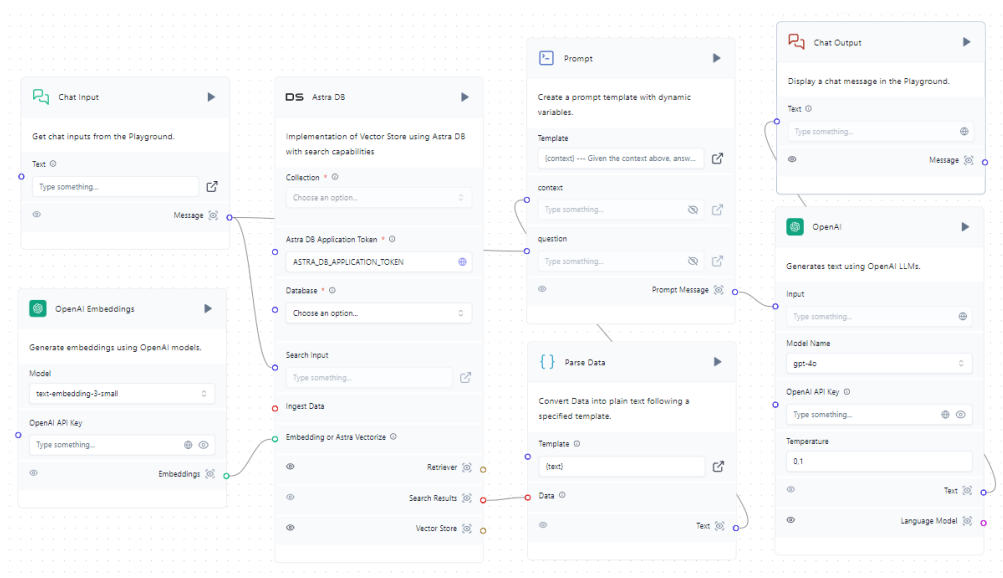


Figura 3.2: RAG mostrado en la herramienta Langflow usando <https://astra.datastax.com>

### Uso de RAGs en la aplicación de este TFG

La utilidad de los *Retrieval Augmented Generation (RAG)* en aplicaciones es muy amplia. La más habitual se usa para conseguir un chatbot de empresas que sirvan como atención al cliente. En nuestro caso se alimenta a la base de datos con embeddings la información actualizada de la web usando agentes que serán explicados a continuación, esta información funciona como una entrada de datos de un sistema RAG para la mayor comprensión del mundo que le rodea al modelo. De esta manera y con un juego de prompts **se obtienen los mejores resultados posibles** que serán después tratados por la aplicación móvil para mostrar dicha información al usuario.

### 3.3. Agentes

La información que alimenta a los RAG puede ser un archivo de texto con información general de un tema sin embargo hay veces en los que la información no está físicamente en un archivo y se tiene que obtener a través de agentes. Estas múltiples herramientas pueden ser vistas como aplicaciones que alimentarán al modelo con un conjunto de herramientas tales como motores de búsqueda, bases de datos, páginas web, etc. Una vez provisto

con esta información el modelo es capaz de razonar acerca de las acciones que debe cumplir para obtener el mejor resultado.

### Uso de Agentes en este TFG

Se utilizan varios con el fin de obtener a través de la web información actualizada de los puntos de interés de los lugares que se van a visitar.

Las referencias se incluyen en el texto usando cite [7]. Para citar webs, artículos o libros [3], si se desean citar más de uno en el mismo lugar [1, 3].

## 3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

**Primer item** más información sobre el primer item.

**Segundo item** más información sobre el segundo item.

■

## 3.5. Tablas

Igualmente se pueden usar los comandos específicos de L<sup>A</sup>T<sub>E</sub>X o bien usar alguno de los comandos de la plantilla.

Algunos conceptos teóricos de L<sup>A</sup>T<sub>E</sub>X <sup>1</sup>.

---

<sup>1</sup>Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto





---

## 4. Técnicas y herramientas

---

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

### 4.1. LangChain

En el último año LangChain se ha establecido como uno de los marcos de trabajo más populares del mercado. Esta herramienta multiusos aúna aplicaciones tan necesarias para el mundo de los *Modelos de Lenguaje a Gran Escala* (LLM) como pueden ser base de datos de vectores, memoria, prompts, herramientas, agentes como ya hemos visto en la sección 3.3 y cadenas de pensamiento (de así su nombre chain). En el prototipo de prompting de este TFG se puede ver el anidamiento de componentes como son estas cadenas para obtener la mejor entrada posible al modelo y obtener la mejor salida posible, estas cadenas pueden unir componentes como prompts, retrievers, processors, tools o incluso otras cadenas para procesos más complejos. Con todo ello LangChain supone una manera de combinar el poder de los LLM con la lógica de cualquier aplicación.



---

## 5. Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

En este apartado se justificarán las decisiones tomadas acerca del desarrollo de la aplicación, tanto en la construcción de los prototipos como en la generación de la aplicación móvil, sus herramientas y documentación.

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del

código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

## 5.1. Elección de agentes

A la hora de obtener información que procesar por el método **RAG** se valoraron muchas fuentes de datos. Uno de los orígenes de datos de información turística favoritos de los usuarios es Tripadvisor. Contar con la información actualizada de este gigante turístico suponía un gran aliciente. Sin embargo se desestimó su uso por varias razones: la información se podía obtener a través del método scraping o webscraping que toma la información en bruto de la página web y se podía postprocesar, sin embargo, dicha práctica incumpliría los Terminos de Uso del Servicio, ya que Tripadvisor usa un acceso a través de API para obtener la información de su base de datos. Una vez más y cómo ya pasara con ciertos servicios de Google, son gratuitos en un principio pero depende de su uso. En primer lugar se necesita una forma de pago para poder empezar a usar el servicio y su utilización si sobrepasa las 5.000 peticiones al mes incurriría en gastos al desarrollador. En este caso y como pasara con Google y OSM se buscó alternativas que funcionaran de manera análoga a Tripadvisor para nutrir el **RAG**.

## 5.2. Elección de servicios Google sobre tecnología OSM

Desde un comienzo el proyecto quería que usara **código abierto** pues comprendo que su filosofía se alinea mejor con los valores aprendidos en la Universidad donde se promociona el uso de herramientas que no supongan un coste para el alumno, se fomenta su uso evitando la posible discriminación económica y una forma de trabajar colaborativa.

De manera renuente decidí cambiar los servicios necesarios para la visualización y gestión de marcadores y rutas a los establecidos por Google. Los motivos que propiciaron este gran cambio fueron los siguientes:

- **Soporte de un gigante tecnológico:** las herramientas de código abierto aunque algunas tienen un gran seguimiento por la comunidad no pueden competir con la documentación, ejemplos de desarrolladores, tecnología de uso, etc. de una potencia como Google.
- **Integración:** Flutter forma parte de Google, lo que supone una integración nativa que hace de su funcionamiento y robustez una de las herramientas usadas.
- Versiones de complementos creadas por terceros con mínimo soporte: cuando se buscaba una solución de rutas como **OSRM: Open Street Route Map**, las peticiones GET había que procesarlas y mapearlas manualmente o usar paquetes de terceros que ya lo habían diseñado, sin embargo estos paquetes aunque faciliten el trabajo suponen también un riesgo ya que son más susceptibles a dejar de funcionar con los cambios de versiones o con la falta de mantenimiento por parte de sus desarrolladores. Un escenario difícil de contemplar si hablamos de interceptores, una clase nativa del paquete **dio** un **"flutter favorite"** (así marca Flutter en pub.dev los paquetes favoritos) con soporte nativo que es la manera en que se ha decidido trabajar desde soluciones Google.

### 5.3. Elección de servicios Geocoding MapBox sobre servicios Google

Los servicios de geocoding son herramientas que permiten convertir direcciones físicas (como Calle Mayor, Ciudad, País") en coordenadas geográficas (latitud y longitud), y viceversa. Esto es esencial para aplicaciones que requieren localización geográfica precisa, como mapas interactivos, planificación de rutas, análisis espacial, o cualquier función que dependa de ubicaciones específicas y los puntos de interés alrededor de una ubicación. Estos servicios en Google se llaman: "Google Places" el uso de sus tokens o API's suponen un coste desde el momento 0. A diferencia del resto de servicios éstos se empiezan a cobrar hasta los 1000 usos al mes. Para una versión inicial de la aplicación de este **TFG** se decidió usar MapBox, una empresa de mapas con un soporte igual de bueno que Google pero que no incurre gastos en

volúmenes de trabajo bajos o moderados. En etapas de desarrollo posteriores o con una aplicación ya establecida y con un uso elevado de peticiones siempre se puede cambiar a Google Places ya que la implementación como pasaba con el resto de servicios es nativa y por tanto muy sencilla de llevar a cabo, si se quiere conseguir unificación de servicios y costes. Desde el punto de vista de la programación solo cambia la manera de acceder a los datos pues las coordenadas se presentan primero con longitud y después con latitud, es decir, de manera inversa a los servicios Google. La modularidad del código realizado y el tratamiento de interceptores hace que un futuro cambio de servicios y peticiones GET sean fácilmente implementados.

---

## 6. Trabajos relacionados

---

A continuación, se comparan aplicaciones de referencia o similares usadas por usuarios que puedan ser objetivos.

### **Tripadvisor**

Tripadvisor permite a los usuarios planificar y organizar sus viajes, con recomendaciones basadas en reseñas y experiencias de otros viajeros. Se puede planificar un viaje personalizado indicando fechas pero prioriza los tours de pago, lo que hace difícil seleccionar rutas saludables gratuitas. Además establecimientos pueden promocionar en la plataforma para mejorar su visibilidad lo que interfiere con la objetividad cuando se busca un lugar a visitar.

### **Wanderlog**

Wanderlog es una aplicación para la planificación de viajes que simplifica la creación de itinerarios, permitiendo agregar lugares de interés fácilmente. La aplicación tiene infinidad de funcionalidades. Tiene una versión Pro con asistente IA pagando 5 euros al mes. Gratuitamente se pueden enviar 5 mensajes, pero carece de contexto el chat. En la imagen se muestra como da lugares de visita de Salamanca por defecto (que se podría obtener de cualquier LLM usando un prompting sencillo) y luego como cambia y no tiene en contexto la respuesta anterior:

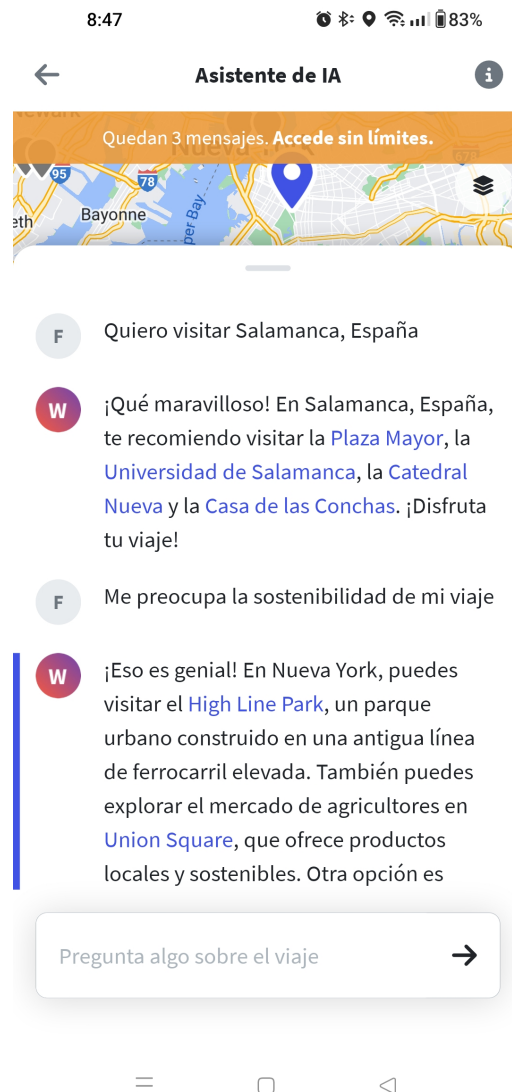


Figura 6.1: Chatbot de Wanderlog

## Visit A City

Visit A City ofrece itinerarios prediseñados para destinos turísticos, permitiendo a los usuarios explorar lugares recomendados según su tiempo disponible. Aunque el uso de la aplicación es gratis, las rutas son todas comerciales y por tanto de pago, aunque existen lo que la aplicación llama planes que son itinerarios generados por usuarios que son gratuitos. Las rutas usan también aplicaciones de terceros.



## Tiqets - Museos y Atracciones

Tiqets es una aplicación que permite comprar entradas para museos y atracciones, ofreciendo guías digitales para planificar visitas. Sólo muestra información de museos, la gestión de mapas es fija, no se puede modificar el zoom y para rutas usa una aplicación por defecto del dispositivo en el que se ejecuta.

## Chatbot - <https://github.com/jrg1013/chatbot>

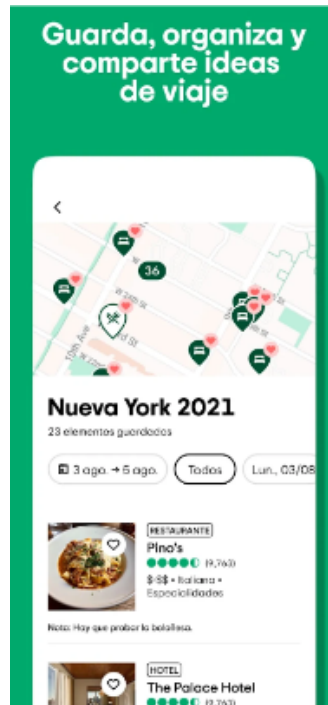
Trabajo Fin de Carrera de José María Redondo Guerra que tuvo como tutores a José Ignacio Santos Martín y Carlos López Nozal.

Este trabajo fue desarrollado usando *Modelos de Lenguaje a Gran Escala (LLM)* para la generación de un chat con un sistema **RAG** para obtener la información de las normas de los **TFG** y así mejorar sus respuestas.

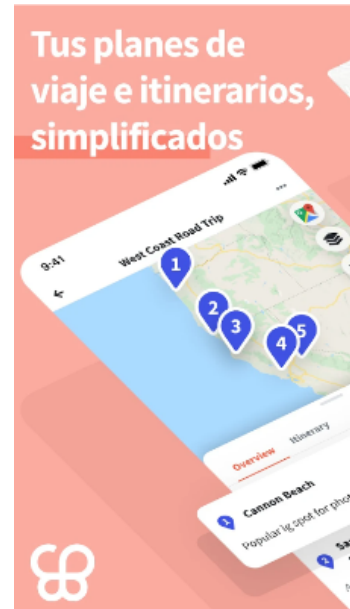
## Visualización de las actividades socioculturales en Castilla y León CULTURALCyL

Trabajo Fin de Carrera de Yanela Lozano Pérez con tutores José Ignacio Santos, Virginia Ahedo y Silvia Díaz. Esta aplicación mostraba información de eventos culturales para lo cual usaba una aplicación móvil con servicios API. Es especialmente interesante como ejemplo de una aplicación móvil con una fuerte característica de usabilidad, ya que se centra en la visualización de mucha información que debe recibir el usuario de manera clara y sencilla.

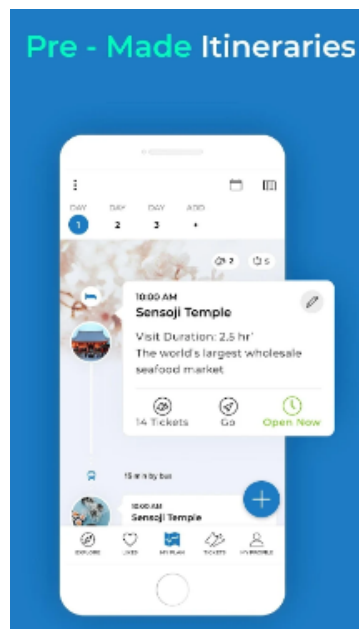
### Tripadvisor



### Wanderlog



### Visit A City



### Tigets - Museos y Atracciones

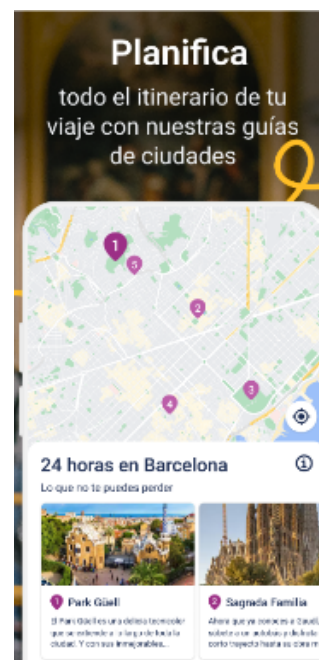


Tabla 6.1: Aplicaciones Similares

Característica	Tripadvisor	Wanderlog	Visit A City	Tiquets	<i>Eco City Tour</i>
Versión de pago	Sí	Sí	No	No	No
Recomendaciones tienen en cuenta factores de sostenibilidad	No	No	No	No	Sí
Modificación dinámica de rutas	Sí	No	No	No	Sí
Intereses de terceros pueden afectar a los resultados	Sí	Sí	No	No	No
Fuente de datos	Propia	Usuarios - LLM	Usuarios	Propia	LLM

Tabla 6.2: Comparación de aplicaciones similares



---

## 7. Conclusiones y Líneas de trabajo futuras

---

### Conclusiones

### Líneas de trabajo futuras

Existen múltiples maneras de expandir y llenar de nuevas funcionalidades a la aplicación llevada a cabo. Por citar algunas que puedan resultar más útiles al usuario:

- **Gamificación:** Recompensas por rutas completadas o distancia recorrida con un medio ecológico.
- **Ratings:** Valorar las rutas permitiendo la búsqueda de los mismos.
- **Mejora en planificador de rutas:** Determinación de la zona de sombra.
- **Multiplataforma:** La aplicación podría beneficiarse de su adaptación a otras plataformas, donde se tendría que tener en cuenta principalmente los permisos de localización. Al utilizar Flutter esta adaptación se podría realizar sobre el mismo código base, facilitando en gran medida su consecución.



---

## Bibliografía

---

- [1] Zachary J Bortolot and Randolph H Wynne. Estimating forest biomass using small footprint lidar data: An individual tree-based approach that incorporates training data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(6):342–360, 2005.
- [2] G.H. Ionescu, D. Firoiu, A.-M. Manda, R. Pîrvu, E. Jianu, and M.-E. Antoniu. Progress towards the 2030 Sustainable Development Goals for EU Urban Communities (SDG11). *Sustainability (Switzerland)*, 16(11), 2024.
- [3] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [4] O. Mitas, R. Badal, M. Verhoeven, K. Verstraten, L. de Graaf, H. Mitsova, W. Weijdemá, and J. Klijs. Tell Me Where to Go: An Experiment in Spreading Visitor Flows in The Netherlands. *International Journal of Environmental Research and Public Health*, 20(8), 2023.
- [5] M.J. Nieuwenhuijsen. Urban and transport planning pathways to carbon neutral, liveable and healthy cities; A review of the current evidence. *Environment International*, 140, 2020.
- [6] J. Vaid. SUSTAINABLE TOURISM: A ROADMAP FOR THE 2030 SUSTAINABLE DEVELOPMENT AGENDA. *Contemporary Studies in Economic and Financial Analysis*, 113B:21–26, 2024.
- [7] Wikipedia. Latex — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=84209252>, 2015. [Internet; descargado 30-septiembre-2015].