



MÓDULO PROYECTO

CFGS Desarrollo de Aplicaciones Multiplataforma
Informática y Comunicaciones

Gestor Documental mediante IA Generativa DocMan

Tutor individual: Jorge Pozo Cata

Tutor colectivo: Cristina Silván Pardo

Año: 2023/2024

Fecha de presentación: 14/06/2024

Nombre y Apellidos: Carmen García Encinas

Email: mcarmen.garenc@educa.jcyl.es

Tabla de contenido

1 Identificación proyecto.....	4
2 Organización de la memoria	4
3 Descripción general del proyecto	4
3.1 Objetivos	4
3.2 Cuestiones metodológicas	5
3.3 Entorno de trabajo.	7
4 Descripción general de la aplicación.	16
4.1 Visión general del sistema.	16
4.2 Descripción breve de métodos, técnicas o arquitecturas (m/t/a) utilizadas.	18
4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha	18
5 Planificación y presupuesto	20
6 Documentación Técnica: análisis, diseño, implementación y pruebas.	21
6.1 Especificación de requisitos.	21
6.2 Diseño del sistema.	22
6.2.1 Diseño de la Base de Datos	22
6.2.2 Diseño de la Interfaz de usuario.	22
6.2.3 Diseño de la Aplicación.	32
6.3 Implementación:	34
6.3.1 Entorno de desarrollo.	34
6.3.2 Estructura del código.	34

6.3.3 Cuestiones de diseño e implementación reseñables.	35
6.4 Pruebas.	37
7 Manuales de usuario	38
7.1 Manual de usuario	38
7.2 Manual de instalación	47
8 Conclusiones y posibles ampliaciones	49
9 Bibliografía	49

1 Identificación proyecto.

A continuación se detallará el Proyecto Final de Curso del FP Desarrollo de Aplicación Multiplataforma. El cual tratará sobre un gestor documental mediante IA generativa.

El proyecto consiste en un sistema avanzado para la gestión y consulta de datos utilizando la base de datos integrada en un RAG y su procesamiento de ficheros. El sistema está diseñado para manejar volúmenes de datos, ofreciendo respuestas rápidas a consultas complejas.

2 Organización de la memoria

1. Identificación del Proyecto: Portada y datos del proyecto.
2. Organización de la Memoria: Enumeración y descripción de los apartados de la memoria.
3. Descripción General del Proyecto: Objetivos, metodología, entorno de trabajo.
4. Descripción General del Producto: Visión del sistema, límites, funcionalidades, usuarios.
5. Planificación y Presupuesto: Cronograma, costes.
6. Documentación Técnica: Análisis, diseño, implementación, pruebas.
7. Manuales de Usuario: Manual de usuario, manual de instalación.
8. Conclusiones y Posibles Ampliaciones: Evaluación del proyecto, futuras mejoras.
9. Bibliografía.
10. Anexos.

3 Descripción general del proyecto

3.1 Objetivos

El objetivo principal del proyecto es desarrollar un gestor documental basado en inteligencia artificial generativa que permita procesar ficheros PDF. Utilizando LlamaIndex y un modelo de lenguaje (LLM) Mistral, el sistema debe ser capaz de responder a preguntas sobre el contenido

de los documentos procesados, así como sobre la información que ya tiene almacenada en la base de datos integrada en LlamaIndex.

Objetivos Secundarios

- Automatizar la extracción de información relevante de documentos PDF.
- Proveer una interfaz de chat para la interacción con los datos extraídos.
- Asegurar la precisión y relevancia de las respuestas generadas por el sistema.
- Asistente de ayuda con consultas rápidas a cerca del funcionamiento de la aplicación.
- Extras de contenido para una navegación más cómoda y completa.
- Apificación de la aplicación.
- Dockerización para realizar el despliegue de la aplicación.

3.2 Cuestiones metodológicas

El proyecto sigue una metodología de desarrollo en cascada, abarcando las siguientes fases:

1. Análisis de Requisitos

En esta fase se identifican y documentan las necesidades y expectativas del proyecto respecto al sistema. Los requisitos funcionales y no funcionales se definen con detalle. Esto incluye la identificación de las funcionalidades esenciales del sistema, las especificaciones técnicas, los objetivos de rendimiento y las restricciones operativas. Al final de esta fase, se produce un documento de requisitos detallado que servirá como base para todas las etapas posteriores del proyecto.

2. Diseño

La fase de diseño implica la creación de los modelos y arquitecturas necesarios para el desarrollo del sistema. Se realizan diagramas de flujo de datos, diagramas de entidad-relación para la base de datos, y se esbozan las interfaces de usuario. Esta fase se divide en dos subfases: diseño de alto nivel (diseño arquitectónico) y diseño detallado. El diseño de alto nivel define la estructura

global del sistema, incluyendo los componentes principales y su interacción. El diseño detallado se enfoca en las especificaciones internas de cada componente, incluyendo algoritmos y estructuras de datos. Los resultados de esta fase son documentos de diseño que guiarán la implementación.

3. Implementación

En esta fase se lleva a cabo la codificación del sistema según los diseños realizados. Se escribe el código fuente utilizando las herramientas y lenguajes de programación seleccionados. Se sigue una estructura modular para facilitar el desarrollo, la prueba y el mantenimiento del código. Durante la implementación, se integran diferentes módulos y componentes, asegurando que trabajen juntos de manera coherente. Además, se realiza una documentación continua del código para asegurar su comprensión y mantenimiento futuros.

4. Pruebas

La fase de pruebas se dedica a verificar y validar que el sistema cumple con los requisitos especificados. Se realizan diferentes tipos de pruebas, incluyendo pruebas unitarias, de integración, de sistema y de aceptación. Las pruebas unitarias evalúan componentes individuales del sistema. Las pruebas de integración verifican la interacción entre diferentes componentes. Las pruebas de sistema aseguran que el sistema completo funciona como se espera, y las pruebas de aceptación validan que el sistema cumple con los requisitos del cliente. Los defectos encontrados se documentan y corrigen iterativamente hasta que el sistema esté listo para el despliegue.

5. Despliegue

En la fase de despliegue, el sistema se instala y se pone en funcionamiento en el entorno de producción. Se preparan y ejecutan scripts de despliegue que configuran la aplicación en contenedores Docker para asegurar un entorno consistente y reproducible. Se proporciona capacitación al usuario final y se crean manuales de usuario y de instalación para facilitar el uso y la gestión del sistema. Esta fase también incluye la configuración de la infraestructura necesaria, la migración de datos (si es necesario) y la realización de pruebas finales para asegurar que el sistema funciona correctamente en su entorno operativo.

3.3 Entorno de trabajo.

Tecnologías.

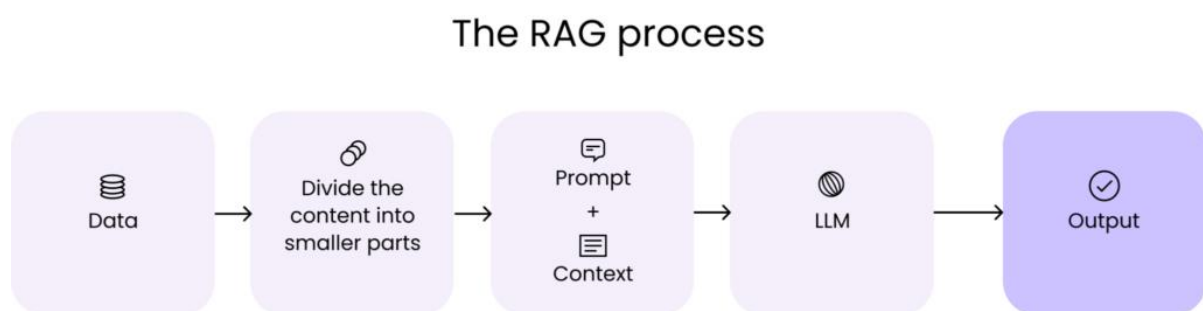
Modelo Cliente-Servidor.

El sistema sigue una arquitectura de modelo cliente-servidor. En esta arquitectura, el cliente (en este caso, la interfaz de usuario creada con Streamlit) solicita servicios y recursos al servidor, que realiza el procesamiento de documentos y genera respuestas basadas en el contenido de los ficheros PDF. Esta separación permite una mejor gestión de recursos y escalabilidad del sistema.

RAG.

RAG, que significa **Retrieval-Augmented Generation**, es una técnica en el ámbito del procesamiento del lenguaje natural que combina la recuperación de información (retrieval) con la generación de texto (generation). Esta técnica se utiliza para mejorar la calidad y la relevancia de las respuestas generadas por un modelo de lenguaje, como los basados en GPT, mediante la integración de información externa relevante en el proceso de generación de texto.

1. Funcionamiento de RAG.

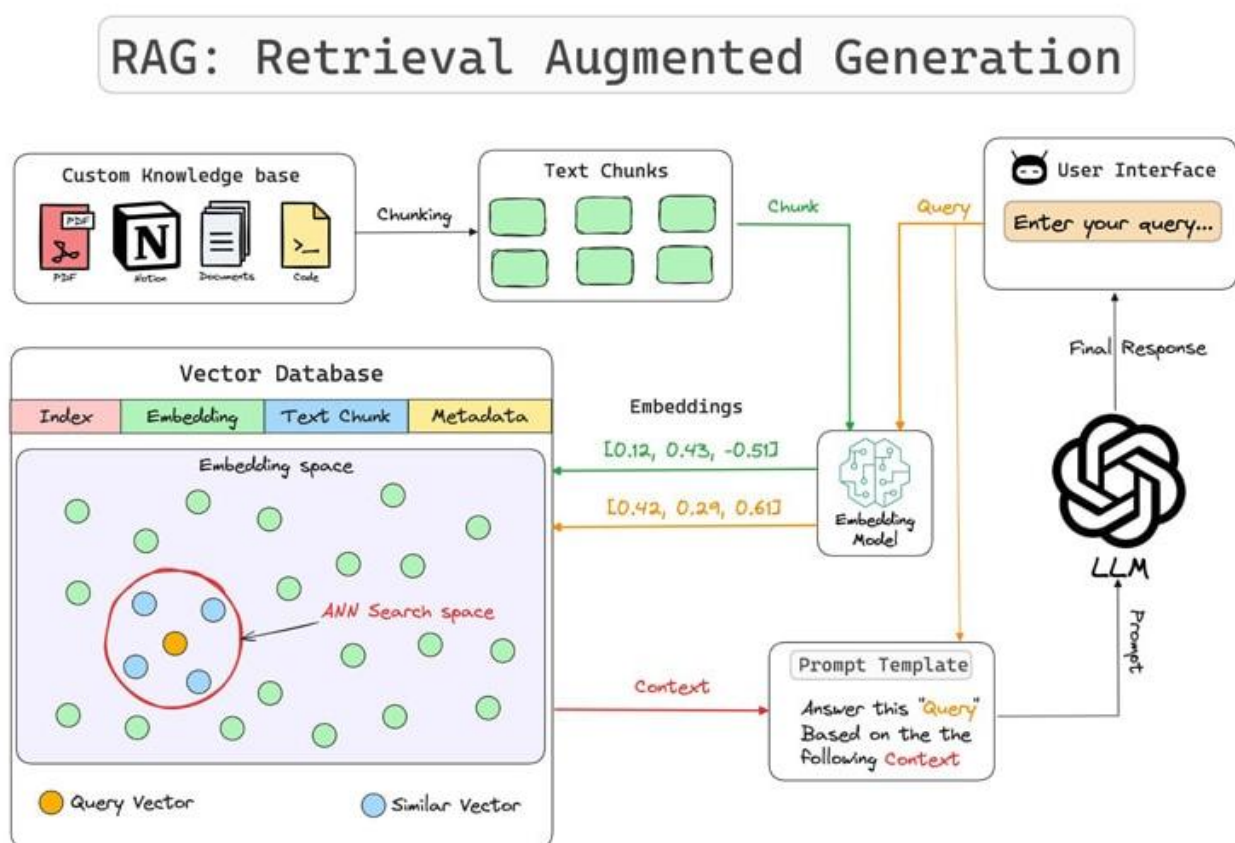


El funcionamiento de RAG se puede desglosar en varios pasos clave:

- **Consulta del Usuario:** El usuario hace una pregunta o solicita información.
- **Recuperación de Información:** Un componente de recuperación de información (como un motor de búsqueda o una base de datos) se utiliza para buscar documentos relevantes que

puedan contener la respuesta o información relacionada con la consulta del usuario. Estos documentos pueden provenir de una variedad de fuentes, como bases de datos, sitios web, artículos científicos, etc.

- **Incorporación de Contexto:** Los documentos recuperados se utilizan como contexto adicional. Este contexto se integra con la consulta original del usuario.
- **Generación de Respuesta:** Un modelo generativo (como un modelo basado en GPT) toma la consulta del usuario junto con el contexto proporcionado por los documentos recuperados y genera una respuesta más informada y precisa.



Para este proyecto se ha escogido la utilización de LlamaIndex, es una biblioteca utilizada para la extracción y procesamiento de información de documentos PDF. Esta herramienta permite convertir documentos en índices vectorizados que facilitan la búsqueda y recuperación de información relevante de manera eficiente. LlamaIndex se encarga de procesar los documentos cargados por los usuarios y extraer la información necesaria para responder a las consultas.

LLM.

Un modelo de lenguaje largo (Large Language Model, LLM) es un tipo de modelo de inteligencia artificial entrenado para comprender y generar texto en lenguaje natural. Estos modelos utilizan arquitecturas avanzadas de redes neuronales, como los transformadores, y son entrenados con enormes cantidades de datos textuales.



Los LLMs pueden realizar una variedad de tareas de procesamiento del lenguaje natural (NLP), incluyendo traducción, resumen, generación de texto, preguntas y respuestas, y más.

1. Funcionamiento de un Modelo LLM

El funcionamiento de un modelo LLM se puede desglosar en varias etapas:

Entrenamiento:

- **Datos:** Se recopilan grandes volúmenes de texto de diversas fuentes (libros, artículos, sitios web, etc.).
- **Preprocesamiento:** Los datos textuales se preprocesan para eliminar ruido y convertirlos en un formato adecuado para el modelo.
- **Arquitectura del Modelo:** Se utiliza una arquitectura de red neuronal avanzada, como un transformador (por ejemplo, GPT-3, BERT).
- **Entrenamiento:** El modelo se entrena mediante aprendizaje supervisado o no supervisado, ajustando los pesos de la red para minimizar la pérdida en la tarea de predicción de palabras.

Inferencia:

- **Entrada:** El usuario proporciona una entrada de texto (consulta o prompt).
- **Tokenización:** La entrada se convierte en una secuencia de tokens (representaciones numéricas de las palabras).

- **Procesamiento:** La secuencia de tokens se pasa a través de la red neuronal, que realiza cálculos en cada capa para generar una representación interna del texto.
- **Generación:** El modelo genera una secuencia de tokens como salida, que se decodifica de nuevo en texto legible.

Para este proyecto se ha optado por la utilización de un modelo LLM de Mistral para la generación de respuestas. Concretamente “**Mixtral 8x7B**” Mistral es un modelo de lenguaje avanzado utilizado para generar respuestas basadas en las consultas de los usuarios. Este modelo emplea técnicas de procesamiento de lenguaje natural (PLN) para interpretar las preguntas y proporcionar respuestas precisas utilizando la información extraída por LlamaIndex. La integración de Mistral permite ofrecer una experiencia de usuario más intuitiva y eficiente. Más adelante se detallará más a cerca de este tipo de modelo.

Lenguaje de Programación

Python (versión 3.9)

Python es el lenguaje principal utilizado para el desarrollo del sistema. Su versatilidad, amplia gama de bibliotecas y facilidad de uso lo hacen ideal para proyectos de procesamiento de lenguaje natural y desarrollo web. Python es un lenguaje de programación de alto nivel y de propósito general, conocido por su legibilidad y capacidad para manejar tareas complejas con relativamente poco código.

Frameworks

Streamlit

Streamlit es una plataforma de desarrollo web que permite crear aplicaciones interactivas rápidamente. En este proyecto, Streamlit se utiliza para desarrollar la interfaz de usuario, proporcionando una forma sencilla y eficaz para que los usuarios carguen documentos PDF, realicen consultas y reciban respuestas. La facilidad de uso de Streamlit permite una rápida iteración y desarrollo de interfaces amigables para el usuario.

FastAPI

FastAPI es un marco web moderno, rápido (de alto rendimiento) para crear API con Python basado en sugerencias de tipo estándar de Python.

Las características clave son:

- **Rápido** : Muy alto rendimiento, a la par de NodeJS y Go (gracias a Starlette y Pydantic). Uno de los frameworks Python más rápidos disponibles .
- **Rápido para codificar** : aumente la velocidad para desarrollar funciones entre un 200% y un 300%.
- **Menos errores** : reduce aproximadamente el 40% de los errores inducidos por humanos (desarrolladores).
- **Intuitivo** : gran soporte para el editor. Finalización por todas partes. Menos tiempo de depuración.
- **Fácil** : Diseñado para ser fácil de usar y aprender. Menos tiempo leyendo documentos.
- **Breve** : Minimiza la duplicación de código. Múltiples características de cada declaración de parámetro. Menos errores.
- **Robusto** : obtenga código listo para producción. Con documentación interactiva automática.
- **Basado en estándares** : basado en (y totalmente compatible con) los estándares abiertos para API:API abierta(anteriormente conocido como Swagger) yEsquema JSON.

Uvicorn

Uvicorn es un servidor ASGI (Asynchronous Server Gateway Interface) ultrarrápido para Python. Es utilizado principalmente para servir aplicaciones web escritas con frameworks como FastAPI.

Las características clave de Uvicorn incluyen:

- **Rápido y eficiente**: Basado en uvloop y httptools, Uvicorn ofrece un rendimiento excelente para aplicaciones web asíncronas.

- **Asincrónico:** Soporte completo para operaciones asíncronas, lo que permite manejar múltiples solicitudes simultáneamente sin bloquear el servidor.
- **Ligero:** Diseñado para ser liviano y de fácil configuración, lo que facilita su integración en proyectos de cualquier escala.
- **Compatible:** Totalmente compatible con la especificación ASGI, lo que lo hace ideal para trabajar con frameworks modernos de Python como FastAPI y Starlette.

En este proyecto, Uvicorn se utiliza para servir la API creada con FastAPI, proporcionando un entorno de servidor rápido y eficiente para manejar las solicitudes de los usuarios.

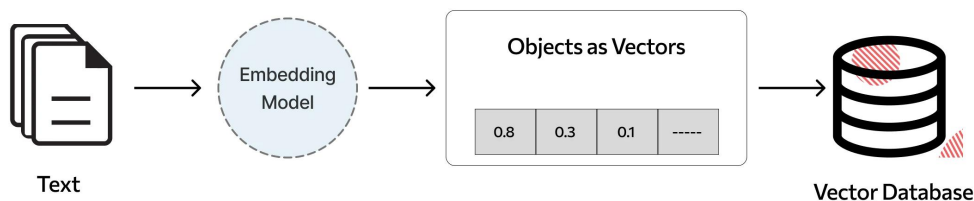
LlamaIndex

LlamaIndex es la biblioteca utilizada para el procesamiento de documentos en el proyecto. Permite convertir documentos PDF en índices vectorizados, facilitando la búsqueda y recuperación de información relevante. Esta biblioteca es crucial para el funcionamiento del gestor documental, ya que proporciona las herramientas necesarias para manejar grandes volúmenes de datos de manera eficiente.

Una de sus características clave es la capacidad de crear una base de datos integrada, donde se pueden almacenar, indexar y recuperar documentos de manera eficiente para mejorar la calidad de las respuestas generadas por modelos de lenguaje grandes (LLMs).

La base de datos integrada de LlamaIndex es una base de datos vectorial. Es esencialmente un índice que organiza y almacena documentos para permitir una recuperación rápida y relevante. Este índice se puede utilizar para realizar búsquedas y consultas eficientes sobre grandes volúmenes de datos textuales.

2. Funcionamiento de la base de datos vectorial.



- **Documentos:** Los documentos son las unidades básicas de información almacenadas en el índice. Un documento puede ser cualquier fragmento de texto, como un artículo, una página web, un libro, etc. En LlamaIndex, cada documento se encapsula en una instancia de la clase **Document**.
- **Índice de Vectores:** LlamaIndex utiliza técnicas de indexación basada en vectores para representar el contenido de los documentos en un espacio vectorial. Cada documento se convierte en un vector utilizando técnicas de **embeddings**, que son representaciones numéricas de texto en un espacio de alta dimensión. Los índices de vectores permiten búsquedas eficientes mediante la comparación de similitudes entre vectores.
- **Creación del Índice:** Para crear un índice, se procesan los documentos y se generan sus representaciones vectoriales. Estos vectores se almacenan en una estructura de datos que permite búsquedas rápidas.
- **Consulta y Recuperación:** Cuando se realiza una consulta, esta también se convierte en un vector. El sistema compara este vector de consulta con los vectores de los documentos almacenados para encontrar los más similares. Los documentos más relevantes se recuperan y se utilizan como contexto adicional para mejorar la generación de respuestas.

Pandas

Pandas es una biblioteca de Python especializada en la manipulación y análisis de datos. Proporciona estructuras de datos rápidas y flexibles, como DataFrame, que permiten una fácil manipulación de datos tabulares. En este proyecto, Pandas se utiliza para procesar y analizar los datos extraídos de los documentos CSV, facilitando la transformación y preparación de los datos para su posterior uso en las consultas y generación de respuestas en el machine learning usado para el chatbot de ayuda.

XGBoost

XGBoost es una biblioteca optimizada de gradient boosting que proporciona una implementación eficiente y flexible para tareas de clasificación y regresión. Es conocido por su rendimiento superior en competiciones de aprendizaje automático y se utiliza ampliamente en la

industria. XGBoost soporta paralelización y optimizaciones que mejoran significativamente la velocidad y eficiencia del modelo.

Scikit-learn

scikit-learn es una biblioteca de aprendizaje automático en Python que proporciona herramientas simples y eficientes para análisis de datos y modelado predictivo. Es compatible con una variedad de algoritmos de clasificación, regresión y clustering, y es ampliamente utilizado debido a su facilidad de uso y extensa documentación.

Transformers

Transformers es una biblioteca desarrollada por Hugging Face que permite el uso de modelos de procesamiento del lenguaje natural (NLP) basados en la arquitectura Transformer. Esta biblioteca soporta una amplia gama de tareas de NLP, como traducción, resumen, y generación de texto. En este proyecto, Transformers se puede utilizar para mejorar la generación de respuestas basadas en el contexto de los documentos procesados por LlamaIndex.

Con la inclusión de estos frameworks, la aplicación DocMan se beneficia de herramientas avanzadas para la creación de interfaces de usuario interactivas, el desarrollo de API rápidas y seguras, y la implementación de técnicas avanzadas de procesamiento de documentos y generación de respuestas. Además, con el soporte de bibliotecas de aprendizaje automático y procesamiento del lenguaje natural, así como herramientas de manipulación de datos, la aplicación puede ofrecer respuestas más precisas y relevantes a las consultas de los usuarios.

Herramientas



Hugging Face

Hugging Face es una plataforma integral para trabajar con modelos de procesamiento del lenguaje natural, proporcionando herramientas poderosas y accesibles para desarrolladores y científicos de datos. La biblioteca **Transformers**, junto con **Datasets**, el Model Hub y Spaces, hace que el desarrollo, la implementación y la experimentación con modelos de NLP sean más eficientes y accesibles.

- **Biblioteca Transformers:** Es una biblioteca de código abierto que proporciona una amplia gama de modelos de aprendizaje profundo preentrenados para tareas de NLP, como generación de texto, clasificación, traducción, y más. Permite a los desarrolladores cargar y usar modelos preentrenados con facilidad, así como entrenar nuevos modelos o ajustar modelos existentes.
- **Datasets:** Hugging Face también ofrece la biblioteca Datasets, que permite a los usuarios acceder y trabajar con una amplia variedad de conjuntos de datos para NLP de manera eficiente.
- **Model Hub:** Es un repositorio centralizado donde los usuarios pueden alojar, compartir y descubrir modelos de IA preentrenados. Proporciona una interfaz web y API para facilitar el acceso y la implementación de modelos.
- **Spaces:** Hugging Face Spaces es una plataforma que permite a los desarrolladores desplegar aplicaciones web interactivas para modelos de machine learning utilizando tecnologías como Gradio y Streamlit.
- **Inferencia como Servicio:** Hugging Face también ofrece servicios de inferencia en la nube, permitiendo a los desarrolladores desplegar modelos y realizar inferencias sin necesidad de manejar la infraestructura. Es esta funcionalidad la que se utiliza en esta aplicación.

Mistral

Mixtral 8x7B es un modelo de lenguaje pre-entrenado utilizado para la generación de respuestas basadas en consultas de los usuarios. Aprovecha técnicas avanzadas de procesamiento de lenguaje natural para entender y responder preguntas con precisión. La integración de este modelo en el proyecto permite una interacción más natural y efectiva entre el usuario y el sistema. Este modelo se caracteriza por la utilización de varios modelos (SMoE), 8 mistrales dispersos (que cada uno ha sido entrenado con conocimientos diferentes) de 7B que se complementan. Utiliza 12,9 mil millones de parámetros activos de un total de 45 mil millones. Habla con fluidez inglés, francés, italiano, alemán, español y domina el código. Además de tener una ventana de contexto de hasta 32K.



Docker

Docker es una plataforma de creación de contenedores que permite empaquetar y desplegar la aplicación de manera consistente en diferentes entornos. Utilizando Docker, se crea una imagen del sistema que incluye todas las dependencias necesarias, asegurando que la aplicación funcione de manera idéntica en cualquier entorno de despliegue. Esto facilita el proceso de instalación y reduce problemas de compatibilidad.



Git

Git es un sistema de control de versiones utilizado para gestionar el código fuente del proyecto. Permite realizar un seguimiento de los cambios, colaborar con otros desarrolladores y mantener un historial de versiones del código. Git es fundamental para asegurar la integridad del código y facilitar la colaboración en equipos de desarrollo.



Visual Studio Code (IDE)

Visual Studio Code (VS Code) es un entorno de desarrollo integrado (IDE) utilizado para escribir y depurar el código del proyecto. VS Code es conocido por su flexibilidad, amplio soporte de extensiones y herramientas integradas para desarrollo en Python y otras tecnologías. Facilita el desarrollo eficiente y la gestión del código, proporcionando un entorno robusto y personalizable para los desarrolladores.

Estas herramientas combinadas permiten que la aplicación no solo sea potente y eficiente, sino también fácil de usar y mantener, proporcionando una solución integral para la gestión documental basada en inteligencia artificial generativa.

4 Descripción general de la aplicación.

4.1 Visión general del sistema.

Visión General del Sistema

El gestor documental es un sistema avanzado basado en inteligencia artificial que procesa ficheros PDF y permite a los usuarios realizar consultas a través de un chat interactivo. Utiliza la información contenida en los documentos y en una base de datos vectorial, proporcionando

respuestas precisas y relevantes. Además da la posibilidad de almacenar dicha información en la base de datos para realizar consultas de esa información en casos posteriores. Así como obtener respuestas en otros idiomas, inglés, y además poder guardar la respuesta obtenida en un fichero de texto.

Por otro lado, también cuenta con la aplicación de la aplicación para poder ser utilizada desde otros servicios.

Límites del Sistema

- Aplicación autónoma desplegada localmente utilizando Docker.
- Dependiente de sistemas operativos compatibles con Docker.

Funcionalidades Básicas

El sistema incorpora las siguientes características clave:

- **Procesamiento de Documentos:** Utiliza LlamaIndex para convertir documentos PDF en índices vectorizados, facilitando la búsqueda y recuperación de información.
- **Base de Datos Vectorial:** La información se almacena en una base de datos vectorizada, optimizando la gestión y recuperación de datos textuales.
- **Generación de Respuestas:** Emplea el modelo de lenguaje Mistral para generar respuestas basadas en el contexto de los documentos procesados.
- **Interfaz de Usuario Interactiva:** Desarrollada con Streamlit, la interfaz es intuitiva y ágil, permitiendo a los usuarios cargar documentos PDF, realizar consultas y recibir respuestas de manera sencilla y eficiente. Facilita la navegación por la aplicación de manera intuitiva además de proporcionar etiquetas de ayuda informando al usuario a lo largo de la aplicación de la funcionalidad de cada componente.
- **Chatbot Personalizado:** Ofrece soporte y ayuda al usuario mediante un chatbot integrado, mejorando la experiencia de usuario y proporcionando asistencia inmediata.

- **Formulario de soporte:** La aplicación cuenta con un formulario para que el usuario en caso de tener una incidencia con la utilización de la aplicación, se envíe un email a soporte técnico.
- **Fornulario de sugerencias del usuario:** La aplicación cuenta con un formulario para que el cliente pueda comunicar sugerencias sobre la aplicación.
- **API:** La aplicación está apificada, de modo que pueda realizarse la conexión a ella desde otros servicios.
- **Despliegue:** Se utiliza Docker para la contenedorización y despliegue de la aplicación, asegurando su ejecución consistente en entornos locales.

4.2 Descripción breve de métodos, técnicas o arquitecturas (m/t/a) utilizadas.

- **Procesamiento de Lenguaje Natural (PLN):** Uso de LlamaIndex y Mistral para la extracción y generación de información.
- **Modelos de Lenguaje:** Utilización de modelos avanzados para asegurar respuestas precisas.
- **Interfaz de Usuario:** Creada utilizando Streamlit para una fácil interacción y visualización.
- **Apificación:** Utilización de FastAPI para la creación de la Api de la aplicación.
- **Machine Learning:** Creación de un chatbot para ayudar al usuario a la utilización de la aplicación y a la actuación ante problemas. Se ha utilizado para ello la técnica de conjunto de algoritmas dentro de la disciplina de una inteligencia artificial, conocido como **machine learning** o aprendizaje automático.

4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha

El despliegue de la aplicación DocMan en un entorno Windows implica varios pasos críticos para asegurar que el sistema funcione correctamente. En esta sección se describen la plataforma tecnológica utilizada, el proceso de instalación y la puesta en marcha de la aplicación en un sistema operativo Windows.

Plataforma Tecnológica

Sistema Operativo

Windows es un sistema operativo ampliamente utilizado en entornos empresariales y personales, conocido por su compatibilidad con una gran variedad de aplicaciones y herramientas de desarrollo.

Contenedorización

Docker Desktop para Windows: Docker Desktop permite a los desarrolladores construir, compartir y ejecutar aplicaciones en contenedores Docker de manera eficiente en un entorno Windows.

Instalación de la Aplicación

El proceso de instalación de DocMan en Windows se realiza en varios pasos, que aseguran la configuración adecuada de todas las dependencias y componentes necesarios. A continuación se detalla el proceso:

Preparación del Entorno

- **Instalación de Docker Desktop:** Descargar e instalar Docker Desktop para Windows desde el sitio oficial de Docker: Docker Desktop for Windows. Durante la instalación, habilitar la integración con WSL 2 (Windows Subsystem for Linux) si es necesario.
- **Instalación de Git:** Descargar e instalar Git para Windows desde el sitio oficial de Git, Git for Windows.
- **Instalación de Visual Studio Code:** Descargar e instalar Visual Studio Code desde el sitio oficial, Visual Studio Code.
- **Instalación de Python:** Descargar e instalar Python 3.9 desde el sitio oficial de Python: Python Downloads. Asegurarse de añadir Python al PATH durante la instalación.

Configuración del Proyecto.

- **Clonar el Repositorio de Docman:** Abrir Git Bash o una terminal de comandos y clonar el repositorio del proyecto.
- **Ejecución de Docker Compose:** Iniciar los servicios definidos en docker-compose.yml.

Puesta en Marcha.

- **Verificación del despliegue.**
- **Acceso a la aplicación:** Una vez que los contenedores estén en funcionamiento, la aplicación estará accesible desde un navegador web en la URL **<http://localhost:8501>**. Esta interfaz permitirá a los usuarios cargar documentos PDF, realizar consultas y recibir respuestas basadas en la información procesada por el sistema. En caso de acceder a la Api, la URL para su utilización es **<http://localhost:8000/docs>**.

Documentación y soporte.

Documentación detallada sobre el uso de la aplicación, incluyendo un manual de usuario y guías de solución de problemas. Además, de soporte para atender cualquier incidencia que los usuarios puedan encontrar durante el uso de la aplicación, mediante un formulario que envía un email a soporte técnico.

5 Planificación y presupuesto

Planificación

- ✓ Mes 1 (Marzo): Análisis de requisitos y diseño inicial.
- ✓ Mes 2 (Abril): Investigación sobre las tecnologías a usar .
- ✓ Mes 3 (Mayo): Desarrollo de la aplicación y pruebas iniciales.
- ✓ Mes 4 (Junio): Despliegue y pruebas finales. Documentación y entrega.

Presupuesto

- **Coste de Desarrollo:** 300 horas a 35€/hora = 10,500€.
- **Coste de Software y Herramientas:** LlamaIndex (gratuito) y Mistral, infraestructuras y licencias para lo que está diseñada la aplicación requeriría de un coste de 2000€ anuales .
- **Coste de Hardware:** 1,500€ (servidor de desarrollo).
- **Coste de Hosting:** N/A (despliegue local).
- **Total:** 14,000€ + mantenimiento y mejoras.

6 Documentación Técnica: análisis, diseño, implementación y pruebas.

6.1 Especificación de requisitos.

1. **Almacenamiento Eficiente de Datos:** Utilización de la base de datos integrada de LlamaIndex para el almacenamiento y gestión eficiente de grandes volúmenes de datos.
2. **Consulta Rápida de Datos:** Soporte para consultas rápidas y eficientes sobre los datos almacenados.
3. **Soporte para Consultas Complejas:** Capacidad para manejar y procesar consultas complejas de manera eficiente.
4. **Interfaz de Usuario Intuitiva:** Diseño de una interfaz amigable y fácil de usar para la carga, consulta y visualización de datos.
5. **Usabilidad:** Diseño centrado en el usuario, asegurando que la aplicación sea fácil de usar y entender.
6. **Chatbot Personalizado:** Ofrece soporte y ayuda al usuario mediante un chatbot integrado, mejorando la experiencia de usuario y proporcionando asistencia inmediata.
7. **Formulario de soporte:** La aplicación cuenta con un formulario para que el usuario en caso de tener una incidencia con la utilización de la aplicación, se envíe un email a soporte técnico.
8. **Formulario de sugerencias del usuario:** La aplicación cuenta con un formulario para que el cliente pueda comunicar sugerencias sobre la aplicación.

9. **API:** La aplicación está apificada, de modo que pueda realizarse la conexión a ella desde otros servicios.
10. **Despliegue:** Se utiliza Docker para la contenedorización y despliegue de la aplicación, asegurando su ejecución consistente en entornos locales.
11. **Mantenimiento y Actualización:** Estructura del código y diseño del sistema que faciliten el mantenimiento y la actualización del software.
12. **Documentación Completa:** Documentación técnica detallada que cubre análisis, diseño, implementación y pruebas del sistema.

6.2 *Diseño del sistema.*

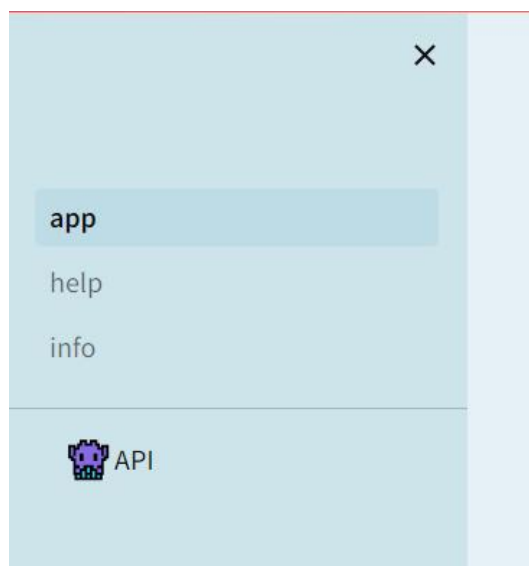
A continuación se detalla el diseño que se ha realizado para la aplicación.

6.2.1 **Diseño de la Base de Datos**

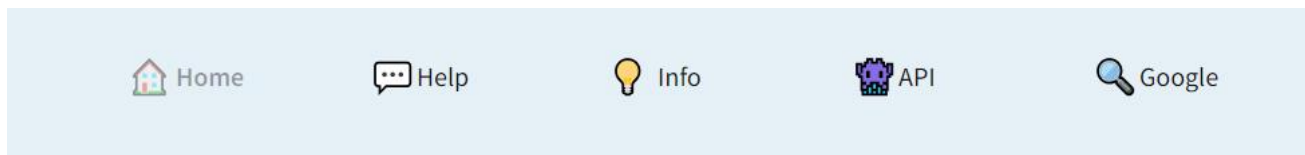
Utilización de la base de datos integrada de LlamaIndex para almacenamiento y consulta eficiente de datos procesados.

6.2.2 **Diseño de la Interfaz de usuario.**

Se ha utilizado Streamlit para la creación de la interfaz de usuario. A lo largo de toda la aplicación existe un menú desplegable que permite al usuario moverse de manera ágil e intuitiva por todas las pantallas.



Además se creado una barra de navegación en la parte superior de las pantallas (app y help) para que de forma visual el usuario pueda ir a la pantalla que necesite de forma más directa. Por otro lado, para guiar mejor al usuario, se difumina la opción de la pantalla donde se encuentra.



Si observamos la imagen veremos que la opción de Home aparece difuminada, esto nos indica que el usuario está posicionado en la pantalla principal.

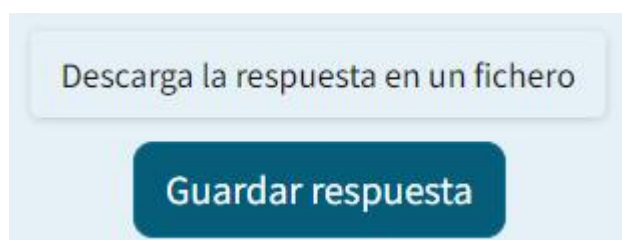
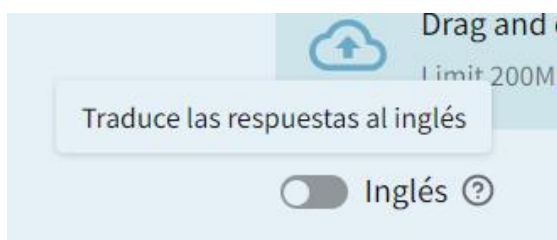
La barra de navegación consta de 5 accesos directos:

- **Home:** Lleva al usuario a la pantalla principal de la aplicación. Donde se encuentra el gestor documental que posteriormente se detallará.
- **Help:** Lleva la usuario a la pantalla de ayuda, donde podrá pedir ayuda de forma inmediata a través de un chatbot para incidencias básicas o preguntas frecuentes de los usuarios.
- **Info:** Lleva al usuario a la pantalla de información, en ella se expone información de la aplicación, se recogen incidencias y sugerencias. Posteriormente se detallará en profundidad.
- **API:** Enlace a la API. Se despliega en FastAPI en el navegador.
- **Google:** Enlace a Google. Da la posibilidad al usuario de realizar búsquedas fuera de la aplicación.

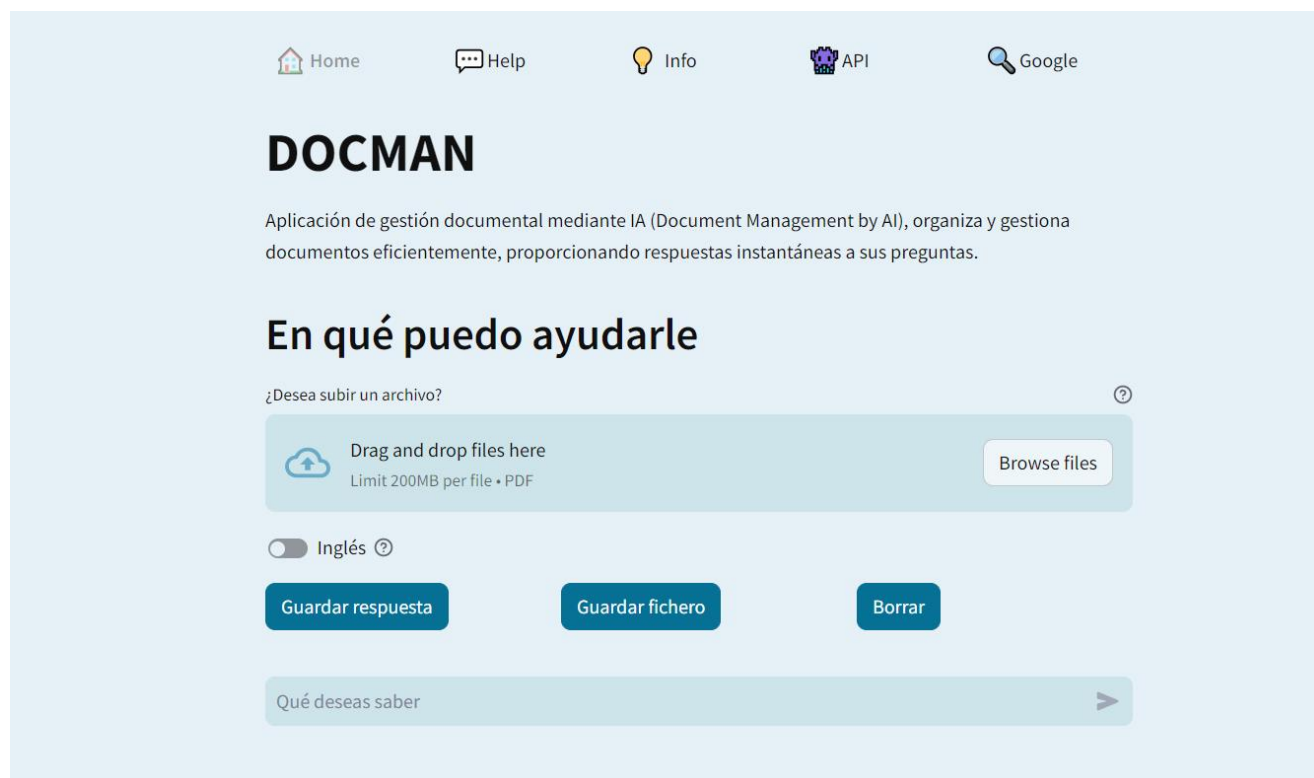
Por otro lado, se han creado etiquetas de ayuda en los botones para que la utilización de la aplicación sea más intuitiva para el usuario.

Estas etiquetas se encuentran a lo largo de toda la aplicación. En los ejemplos que observamos a continuación.

El primer caso se activa cuando el usuario pasa el ratón encima del signo de interrogación. En el segunda caso simplemente es al pasar el ratón por encima de cualquier botón, mostrando la funcionalidad que corresponde en cada caso.



Pantalla Home/App



En la parte superior tiene la barra de navegación que se ha explicado anteriormente y en la parte izquierda aparece el desplegable que también se ha mencionado.

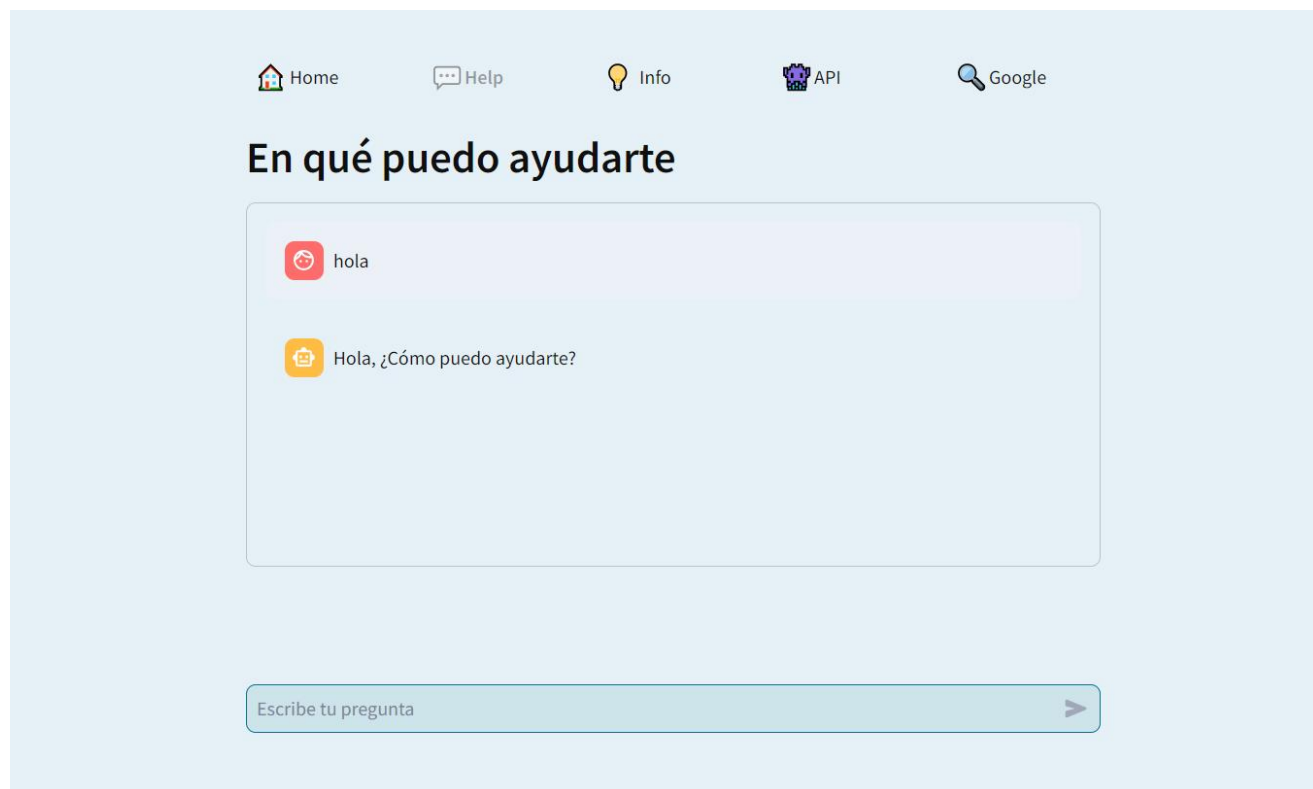
Presenta una breve descripción de la funcionalidad de la aplicación. En la parte posterior, está el botón “Browse File”, mediante el cual al accionarlo aparece una ventana emergente donde el usuario podrá buscar los ficheros que desea cargar para realizar las consultas.

A continuación, aparece un botón de deslizamiento que da la posibilidad al usuario de obtener las respuestas a sus consultas en inglés.

Además de tres botones que dan las funcionalidades de guardar la respuesta obtenida en un fichero, guardar el fichero en la base de datos para realizar futuras consultas y el borrado de la información del fichero.

Finalmente aparece un cuadro de texto donde el usuario debe introducir su consulta. Dicho cuadro ya posee un botón para enviar la consulta.

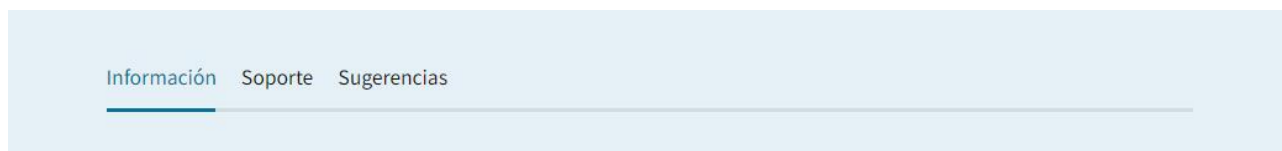
Pantalla Help



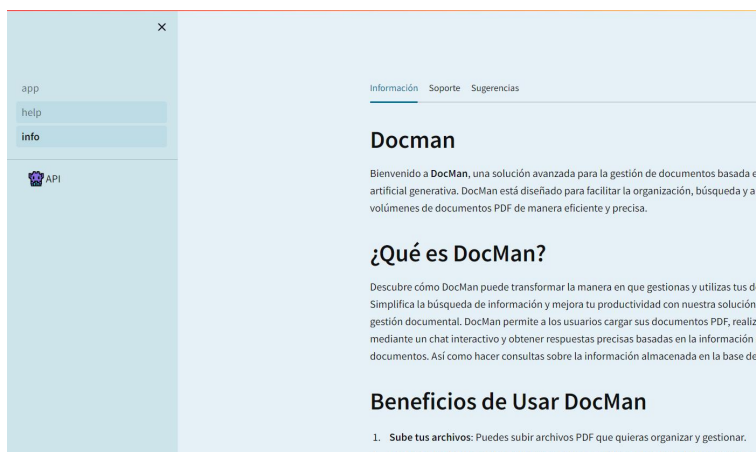
En la parte superior tiene la barra de navegación que se ha explicado anteriormente y en la parte izquierda aparece el desplegable que también se ha mencionado.

Presenta un ChatBot para que el usuario pueda hacer consultas sobre el funcionamiento de la aplicación y sobre posibles preguntas frecuentes que puedan ayudar a que la aplicación se utilice de manera más óptima.

Pantalla Info



En la parte superior tiene la barra de navegación donde permite al usuario moverse por las tres diferentes pantalla: Información, Soporte y Sugerencia.



Apartado de Información.

Información general sobre la aplicación, así como sus funcionalidades.

Docman

Bienvenido a **DocMan**, una solución avanzada para la gestión de documentos basada en inteligencia artificial generativa. DocMan está diseñado para facilitar la organización, búsqueda y análisis de grandes volúmenes de documentos PDF de manera eficiente y precisa.

¿Qué es DocMan?

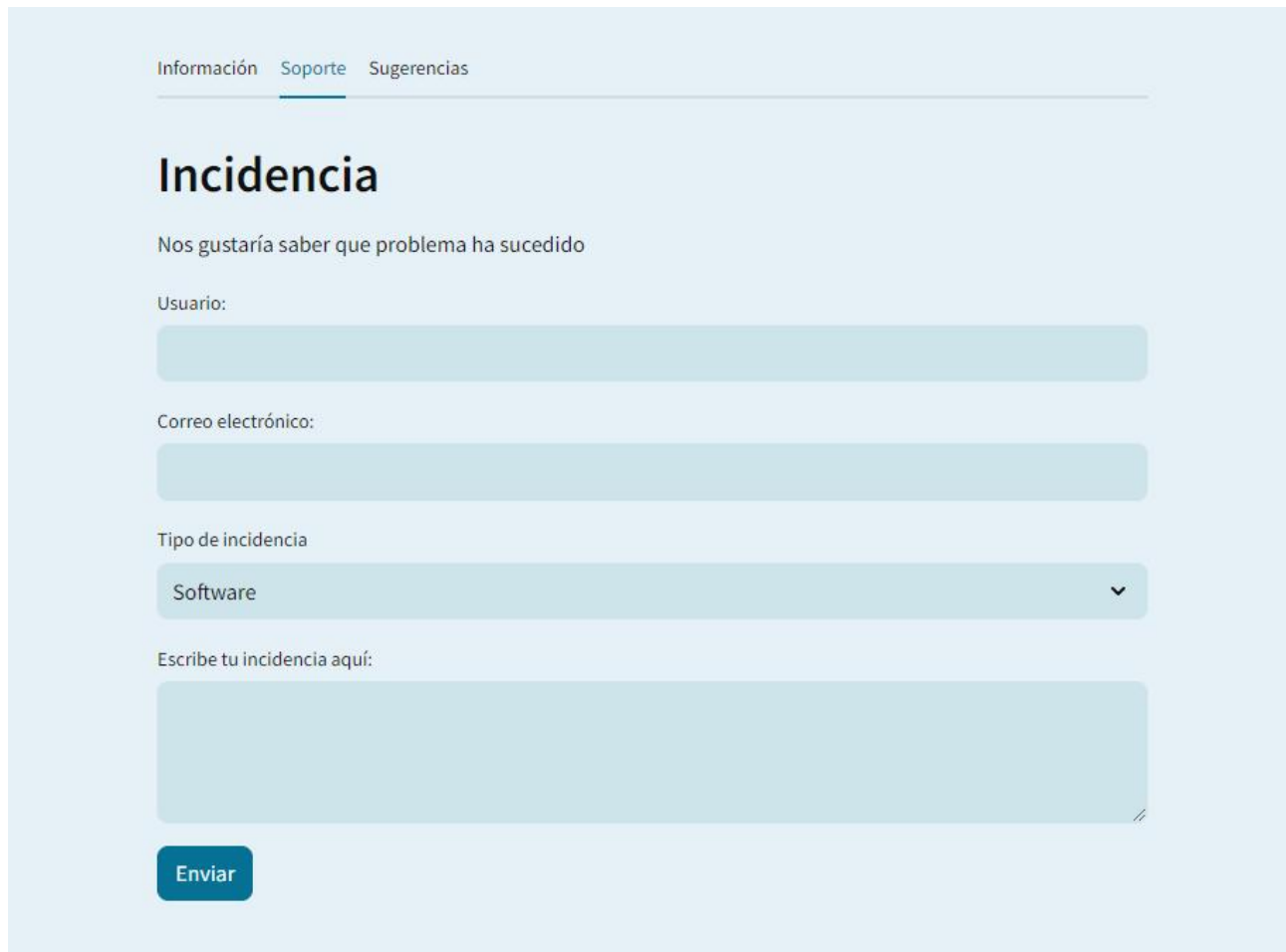
Descubre cómo DocMan puede transformar la manera en que gestionas y utilizas tus documentos. Simplifica la búsqueda de información y mejora tu productividad con nuestra solución inteligente de gestión documental. DocMan permite a los usuarios cargar sus documentos PDF, realizar consultas mediante un chat interactivo y obtener respuestas precisas basadas en la información contenida en esos documentos. Así como hacer consultas sobre la información almacenada en la base de datos.

Beneficios de Usar DocMan

1. **Sube tus archivos:** Puedes subir archivos PDF que quieras organizar y gestionar.
2. **Procesamiento de archivos:** El sistema procesará los archivos subidos para extraer su contenido.
3. **Almacenamiento:** Los documentos se almacenarán en una base de datos vectorial para futuras búsquedas.
4. **Consulta:** Puedes hacer preguntas sobre los documentos almacenados y recibir respuestas instantáneas.
5. **Guardar y eliminar:** Tienes opciones para guardar respuestas o eliminar archivos cargados.

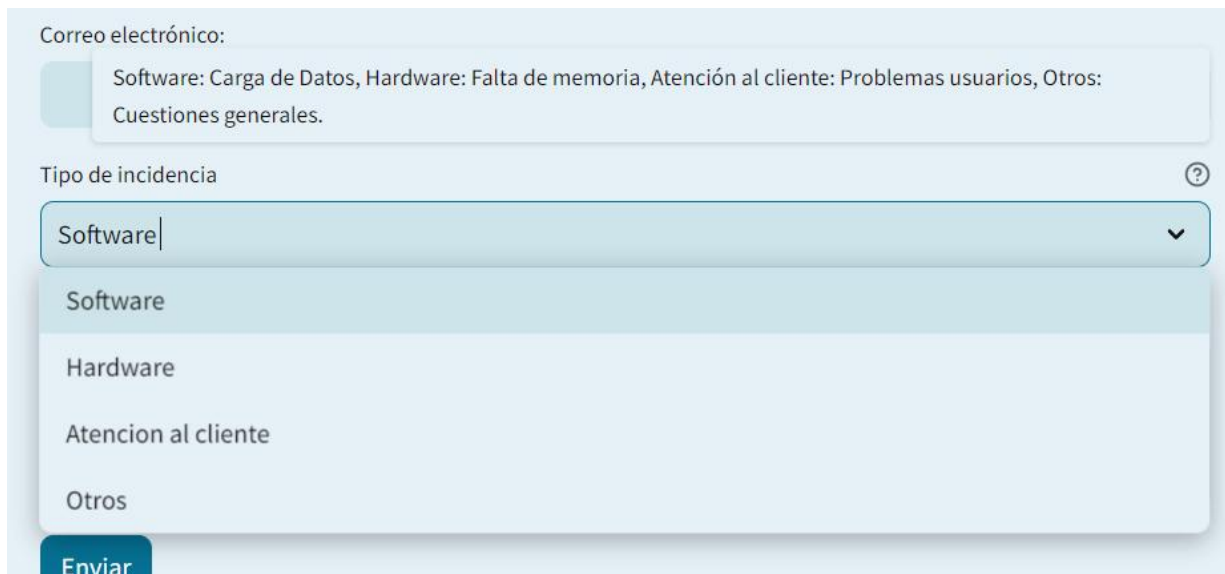


Apartado de Soporte.



The screenshot shows a web form titled 'Incidencia' under a navigation bar with 'Información', 'Soporte', and 'Sugerencias'. The form includes a header 'Nos gustaría saber que problema ha sucedido', followed by input fields for 'Usuario:' and 'Correo electrónico:'. A dropdown menu for 'Tipo de incidencia' is set to 'Software'. Below is a large text area labeled 'Escribe tu incidencia aquí:' and a blue 'Enviar' button.

Presenta un formulario donde el usuario puede introducir la información necesaria sobre su incidencia. Además de un desplegable donde poder enviar la incidencia con mayor precisión.



This close-up shows the 'Correo electrónico:' label and a text box containing 'Software: Carga de Datos, Hardware: Falta de memoria, Atención al cliente: Problemas usuarios, Otros: Cuestiones generales.' Below is the 'Tipo de incidencia' dropdown menu, which is open, showing options: 'Software', 'Hardware', 'Atencion al cliente', and 'Otros'. A blue 'Enviar' button is at the bottom.

Al realizar el envío se validan los datos, devolviendo las siguiente respuestas:

Campos vacíos.

Enviar

Por favor, completa todos los campos.

Correo electrónico no válido.

Enviar

Por favor, introduce un correo electrónico válido.

Envío realizado.

Enviar

Disculpa las molestias. En breve nos pondremos en contacto contigo.

Apartado de Sugerencias.

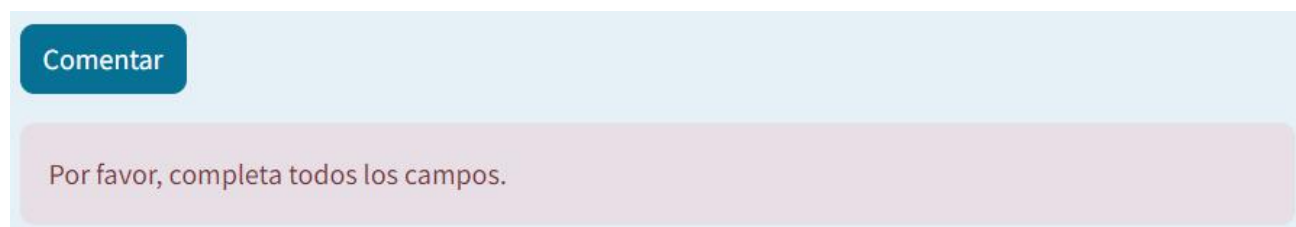
Presenta un formulario donde el usuario puede introducir la sugerencia que crea conveniente. Además de un botón deslizable mediante el cual podrá realizar su suscripción a la plataforma en modo premium.



The screenshot shows a web interface with a light blue background. At the top, there is a navigation bar with three links: 'Información', 'Soporte', and 'Sugerencias', with 'Sugerencias' being the active link. Below the navigation bar, the title 'Sugerencias' is displayed in a large, bold font. Underneath the title, the text 'Nos gustaría conocer tu opinión' is shown. The form consists of three input fields: 'Usuario' (User), 'Email', and a larger text area for the suggestion itself, each with a light blue border. Below the text area, there is a toggle switch labeled 'Deseo suscribirme a la versión Premium'. At the bottom of the form, there is a blue button labeled 'Comentar'.

Al realizar el envío se validan los datos, devolviendo las siguiente respuestas:

Campos vacíos.



The screenshot shows a light blue background with a blue button labeled 'Comentar' at the top left. Below the button, there is a light pink rectangular box containing the text 'Por favor, completa todos los campos.' (Please, complete all fields).

En caso de realizar la suscripción y que el campo del email esté vacío o no tenga un formato correcto.

☒ Deseo suscribirme a la versión Premium

Por favor, Introduzca su correo electrónico.

Correo electrónico no válido.

Comentar

Por favor, introduce un correo electrónico válido.

Suscripción realizada.

☒ Deseo suscribirme a la versión Premium

Gracias por suscribirse! Recibirá un correo electrónico con las instrucciones que debe seguir.

Envío realizado.

En caso de realizarse el envío sin suscripción.

☐ Deseo suscribirme a la versión Premium

Comentar

¡Gracias por tu sugerencia!

En el caso de envío de enviar la sugerencia y a mayores realizar una suscripción, aparecen globos de fiesta para fomentar las suscripciones a la versión premium.



API.

La API consta de dos endpoint. Un get mediante el cual se puede realizar las consultas y un post que da la posibilidad de almacenar documentación en la base de datos.

FastAPI 0.1.0 OAS 3.1

/openapi.json

default

GET /docman Ask

POST /docman Save

Schemas

HTTPValidationError > Expand all object

ValidationError > Expand all object

6.2.3 Diseño de la Aplicación.

El diseño de la aplicación Docman está orientado a ofrecer una experiencia de usuario intuitiva y eficiente, utilizando tecnologías avanzadas para procesar documentos, generar respuestas precisas y facilitar la interacción a través de una interfaz amigable. Este enfoque garantiza que la aplicación no solo sea funcional y escalable, sino también adaptable a futuras mejoras y ampliaciones en funcionalidad y rendimiento. A continuación, se detallan los aspectos clave del diseño:

1. Interfaz de Usuario (UI)

La interfaz de usuario se ha desarrollado utilizando Streamlit, lo que proporciona una experiencia interactiva y amigable para los usuarios finales. Se ha diseñado con los siguientes elementos:

- **Carga de Documentos:** Los usuarios pueden cargar archivos PDF directamente desde sus dispositivos.
- **Consulta a través de Chat:** Incorpora un chat interactivo donde los usuarios pueden realizar consultas utilizando texto natural.
- **Visualización de Resultados:** Muestra las respuestas generadas por el modelo de lenguaje Mistral de manera clara y contextualizada, junto con documentos relevantes recuperados de la base de datos vectorial.

2. Backend - App/Home

Integración con LlamaIndex: Utiliza LlamaIndex para el procesamiento de documentos PDF y la generación de índices vectoriales. Este proceso permite almacenar y recuperar documentos de manera eficiente para su posterior análisis y respuesta.

Este componente realiza las siguientes tareas:

- **Indexación de Documentos:** Genera vectores que representan el contenido semántico de los documentos, facilitando la búsqueda y recuperación rápida de información relevante.

- **Almacenamiento en Base de Datos Vectorial:** Los documentos y sus vectores asociados se almacenan en una base de datos vectorial integrada, optimizando la gestión de grandes volúmenes de datos textuales.

Modelo de Lenguaje (Mistral)

Mistral se utiliza para la generación de respuestas contextuales basadas en el contenido de los documentos procesados. Sus funciones incluyen:

- **Generación de Respuestas:** Utiliza técnicas avanzadas de procesamiento del lenguaje natural para comprender y responder preguntas formuladas por los usuarios a través del chat.
 - **Contextualización:** Las respuestas se generan considerando el contexto proporcionado por los documentos almacenados en la base de datos vectorial, mejorando la relevancia y precisión de las respuestas.
3. Backend ChatBot/Help.
- **Respuestas Automatizadas:** Utiliza modelos de procesamiento del lenguaje natural (NLP) entrenados mediante machine learning para entender las consultas de los usuarios y proporcionar respuestas precisas.
 - **Soporte Interactivo:** Los usuarios pueden escribir sus preguntas y recibir respuestas inmediatas del chatbot. Esto incluye instrucciones sobre cómo utilizar diversas funciones de la aplicación, resolver problemas comunes o proporcionar información adicional sobre el manejo de documentos.

4. Contenedores Docker

Para garantizar la portabilidad y consistencia del sistema, toda la aplicación se ha dockerizado. Este enfoque asegura que la aplicación pueda ser desplegada y ejecutada de manera consistente en diferentes entornos, simplificando el proceso de configuración y mantenimiento.

6.3 Implementación:

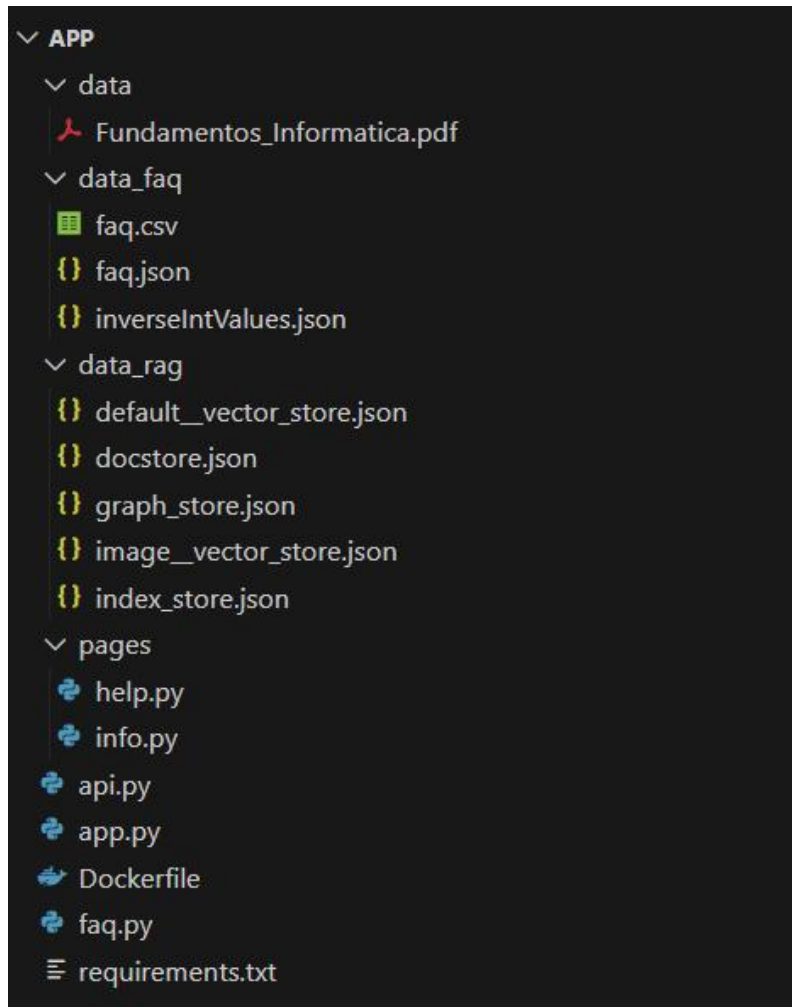
6.3.1 Entorno de desarrollo.

El entorno de desarrollo que se ha utilizado para desarrollar la aplicación es Visual Studio Code.

Visual Studio Code (VS Code) es un entorno de desarrollo integrado (IDE) utilizado para escribir y depurar el código del proyecto. VS Code es conocido por su flexibilidad, amplio soporte de extensiones y herramientas integradas para desarrollo en Python y otras tecnologías. Facilita el desarrollo eficiente y la gestión del código, proporcionando un entorno robusto y personalizable para los desarrolladores.

6.3.2 Estructura del código.

La estructura de la aplicación es la siguiente:



Se han instalado y utilizado las siguientes librerías y frameworks:

```
requirements.txt
1  torch
2  torchvision
3  torchaudio
4  llama-index
5  llama-index-llms-openai-like
6  llama-index-llms-huggingface
7  llama_index-embeddings-huggingface
8  llama-index-llms-azure-openai
9  llama-index-embeddings-azure-openai
10 transformers[torch]
11 huggingface_hub[inference]
12 streamlit
13 streamlit-modal
14 python-dotenv
15 pymupdf
16 sentence-transformers
17 xgboost
18 uvicorn
19 fastapi
20 pandas
```

6.3.3 Cuestiones de diseño e implementación reseñables.

Durante el desarrollo de Docman, se han abordado diversas cuestiones de diseño e implementación que han contribuido significativamente a la funcionalidad y eficiencia del sistema. A continuación, se destacan algunos aspectos relevantes:

Arquitectura Cliente-Servidor: La aplicación se ha diseñado siguiendo una arquitectura cliente-servidor, donde Streamlit actúa como el cliente para la interfaz de usuario interactiva, y FastAPI sirve como el servidor para gestionar las solicitudes y respuestas a través de la API. Actualmente

por falta de tiempo, no están conectados el servicio de la Api con la interfaz de usuario, sino que funcionan de manera independiente.

Integración de Tecnologías Avanzadas: Se ha integrado LlamaIndex para el procesamiento de documentos PDF, aprovechando su capacidad para generar índices vectoriales que facilitan la búsqueda eficiente de información. Además, el modelo de lenguaje Mistral se utiliza para la generación de respuestas contextualizadas basadas en el contenido de los documentos.

Dockerización y Despliegue: La aplicación se dockerizó para asegurar su portabilidad y consistencia en diferentes entornos. Esto ha facilitado el despliegue local del sistema en entornos de desarrollo y pruebas de manera eficiente y sin complicaciones.

Implementación Orientada a Objetos: El código se ha estructurado utilizando principios de programación orientada a objetos en Python, lo cual ha permitido una organización modular y reutilizable del código. Se han definido clases para representar rag, faq y configuraciones del sistema, facilitando la extensibilidad y mantenimiento del código.

Estas consideraciones de diseño e implementación han sido fundamentales para desarrollar un gestor documental robusto, eficiente y escalable, capaz de ofrecer respuestas precisas y relevantes a los usuarios mediante el uso inteligente de tecnologías avanzadas y buenas prácticas de desarrollo de software.

6.4 Pruebas.

A continuación se detallan las pruebas más relevantes, pero se han llevado a cabo todas las pruebas necesarias para el buen funcionamiento de la aplicación.

Funcionalidad	Caso de Prueba	Entrada	Salida
Gestión de Documentos	Subir un documento PDF	Archivo Pdf	Documento almacenado correctamente
	Subir un documento txt	Archivo txt	Mensaje “files are not allowed”
	Borrado ficheros	Fichero	Mensaje “Fichero borrado”
	Consulta información contenida en BBDD	Consulta información contenida.	Respuesta correcta.
	Consulta información no contenida en BBDD.	Consulta información no contenida.	Respuesta no encontrado en contexto.
	Consulta información contenida en fichero.	Consulta información contenida.	Respuesta correcta.
	Consulta información no contenida en fichero ni BBDD.	Consulta información no contenida.	Respuesta no encontrado en contexto.
Interfaz de Usuario	Navegación intuitiva por la interfaz.	Accesos directos, botones, menús.	Acceso fluido a todas las funcionalidades.
Integración con LlamaIndex	Indexación de documentos.	Documento Pdf.	Documento indexado correctamente.
ChatBot	Consulta precisa y clara.	Consulta.	Respuesta correcta.
	Consulta mal redactada y confusa.	Consulta.	Respuesta errónea.

7 Manuales de usuario

7.1 Manual de usuario

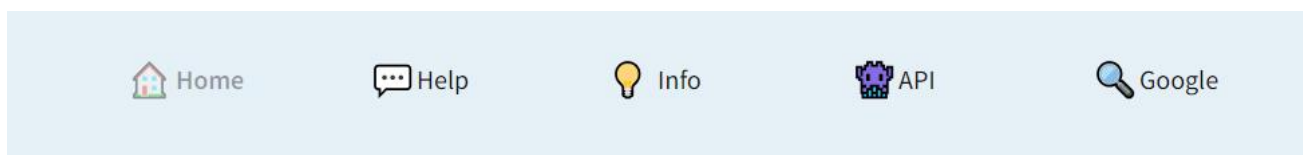
1. Acceso a la Aplicación

Abra su navegador web.

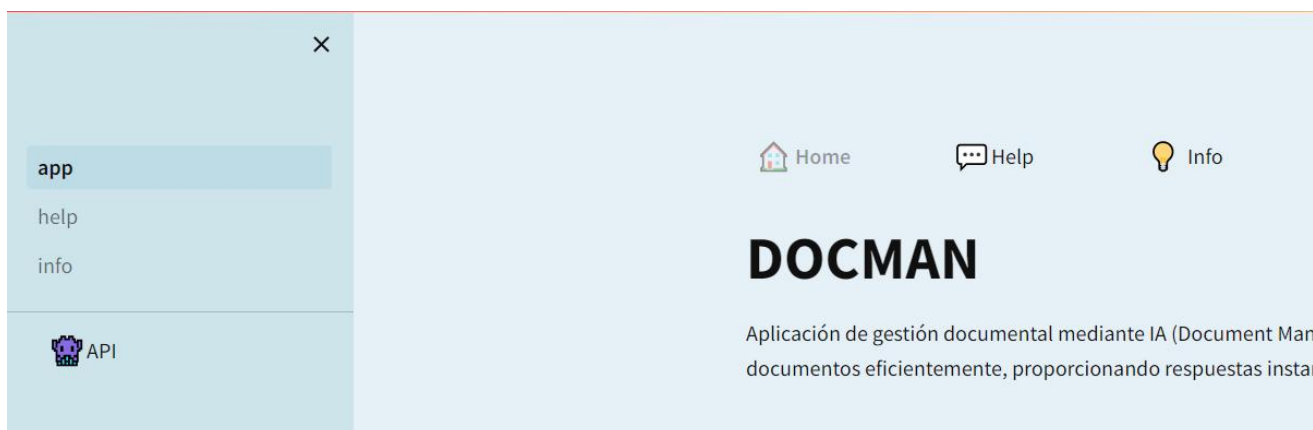
Ingresa la URL proporcionada para acceder a la aplicación: **http://localhost:8501**

2. Navegación General

En la parte superior de la pantalla, encontrará la barra de navegación con las opciones Home, Help, Info, API, y Google.



En la parte izquierda de la pantalla, encontrará un menú desplegable que permite moverse entre las diferentes secciones de la aplicación.

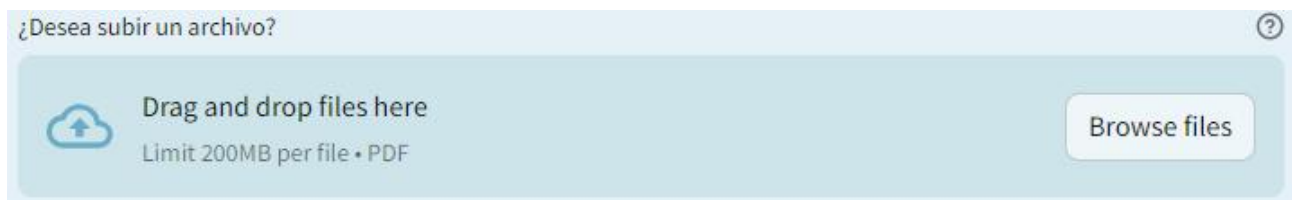


4. Pantalla Home/App



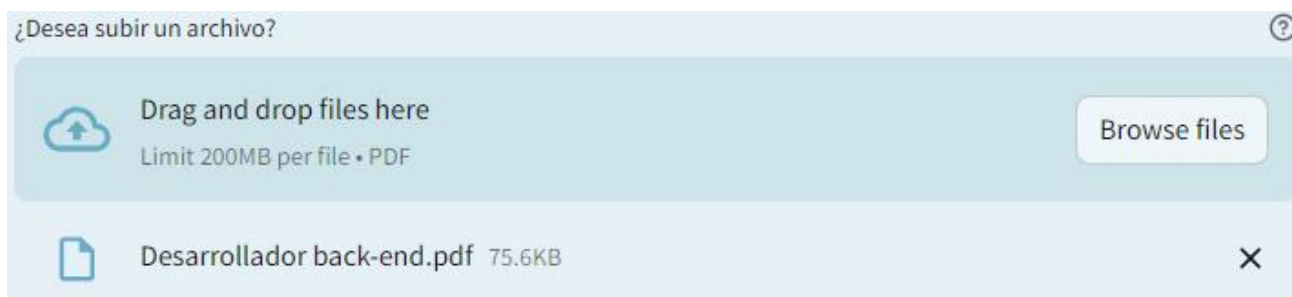
Cargar Archivos:

Haga clic en el botón "Browse File".



Seleccione el archivo que desea cargar desde su ordenador.

Haga clic en "Abrir" para cargar el archivo. Su archivo aparecerá cargado de lo contrario vuelva a realizar las anteriores instrucciones.



Realizar Consultas:

Introduzca su consulta en el cuadro de texto ubicado en la parte inferior de la pantalla.

>

Haga clic en el botón "Enviar" para procesar la consulta.

Opciones de Consulta en inglés:

Utilice el botón de deslizamiento para obtener respuestas en inglés.



Español



Inglés

Los botones adicionales:



Guardar la respuesta obtenida en un archivo.

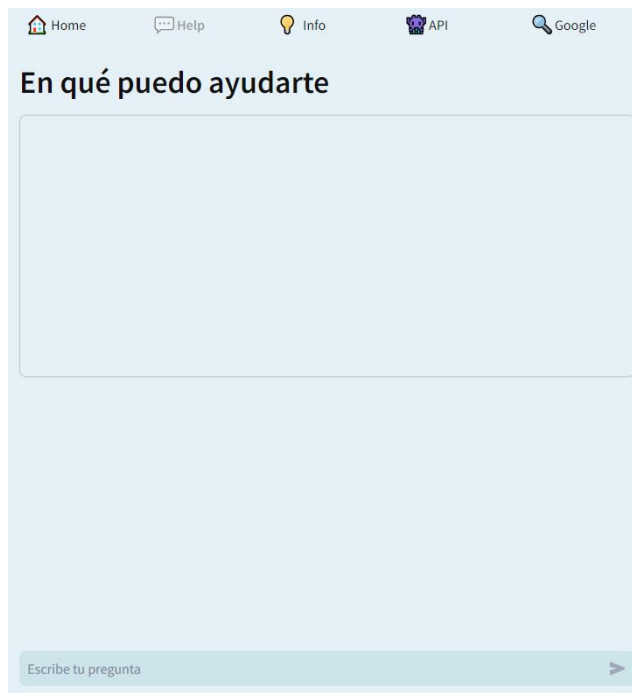


Guardar el archivo en la base de datos para futuras consultas.



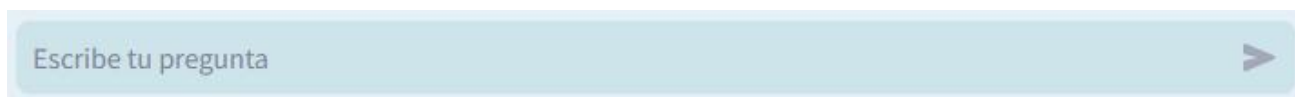
Borrar la información del archivo cargado.

5. Pantalla Help

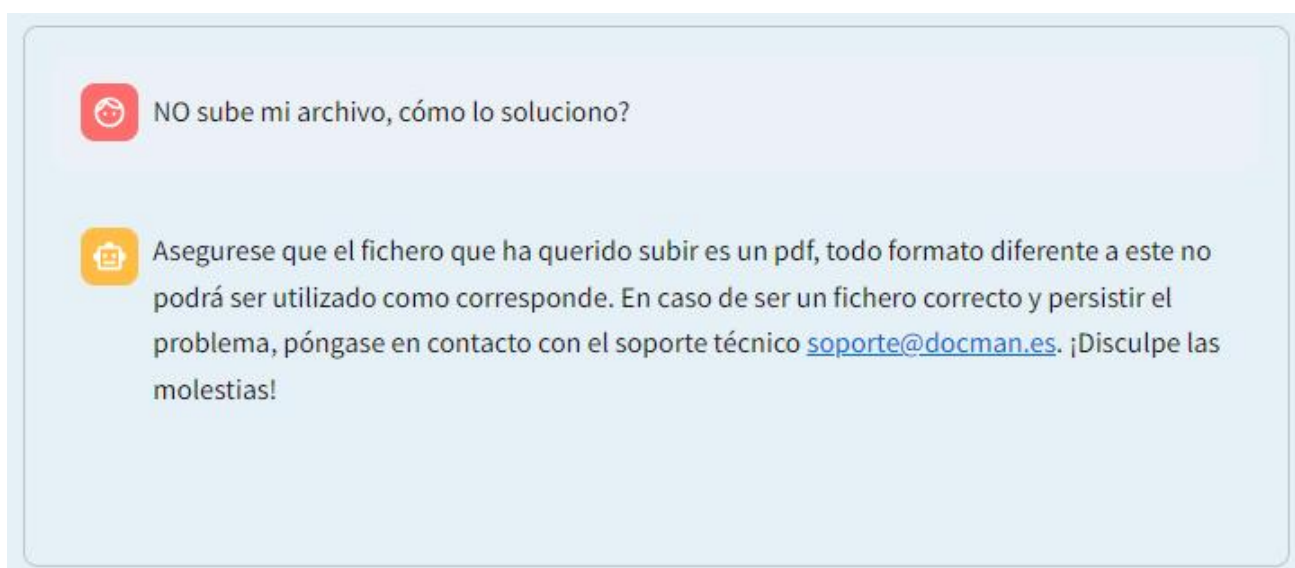


ChatBot:

Escriba su pregunta o problema en el campo de texto del chatbot.

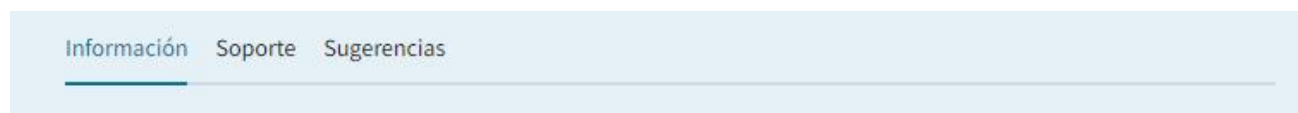


El chatbot responderá con la información o asistencia que necesite.



6. Pantalla Info

En la parte superior de la pantalla, encontrará la barra de navegación con las opciones Información, Soporte y Sugerencias.



Apartado de Información:


Bienvenido a **DocMan**, una solución avanzada para la gestión de documentos basada en inteligencia artificial generativa. DocMan está diseñado para facilitar la organización, búsqueda y análisis de grandes volúmenes de documentos PDF de manera eficiente y precisa.

¿Qué es DocMan?

Descubre cómo DocMan puede transformar la manera en que gestionas y utilizas tus documentos. Simplifica la búsqueda de información y mejora tu productividad con nuestra solución inteligente de gestión documental. DocMan permite a los usuarios cargar sus documentos PDF, realizar consultas mediante un chat interactivo y obtener respuestas precisas basadas en la información contenida en esos documentos. Así como hacer consultas sobre la información almacenada en la base de datos.

Beneficios de Usar DocMan

1. **Sube tus archivos:** Puedes subir archivos PDF que quieras organizar y gestionar.
2. **Procesamiento de archivos:** El sistema procesará los archivos subidos para extraer su contenido.
3. **Almacenamiento:** Los documentos se almacenarán en una base de datos vectorial para futuras búsquedas.
4. **Consulta:** Puedes hacer preguntas sobre los documentos almacenados y recibir respuestas instantáneas.
5. **Guardar y eliminar:** Tienes opciones para guardar respuestas o eliminar archivos cargados.



Consulte la información general sobre la aplicación y sus funcionalidades.

Apartado de Soporte:

Incidencia

Nos gustaría saber que problema ha sucedido

Usuario:

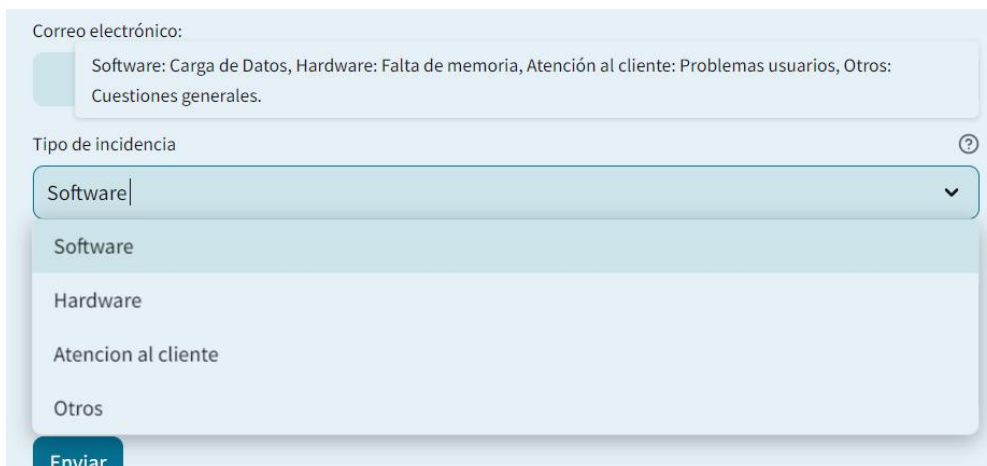
Correo electrónico:

Tipo de incidencia:

Escribe tu incidencia aquí:

Complete el formulario con la información necesaria sobre su incidencia.

Seleccione la categoría de la incidencia en el desplegable.



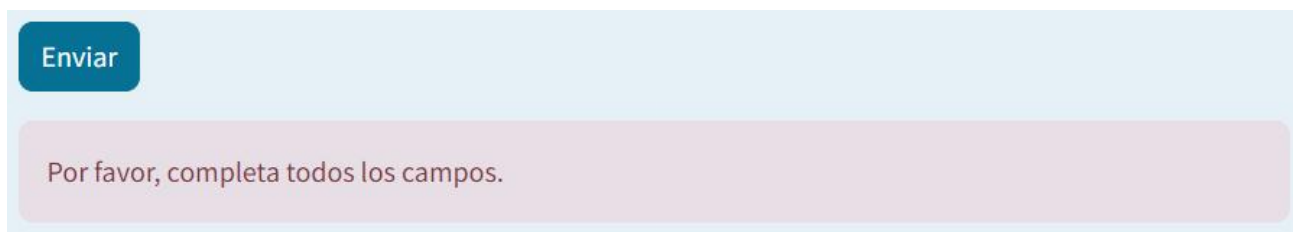
The screenshot shows a form for reporting an incident. At the top, there is a section for 'Correo electrónico:' with a text input field containing the placeholder text: 'Software: Carga de Datos, Hardware: Falta de memoria, Atención al cliente: Problemas usuarios, Otros: Cuestiones generales.' Below this is a section for 'Tipo de incidencia' with a dropdown menu. The dropdown is currently open, showing four options: 'Software', 'Hardware', 'Atencion al cliente', and 'Otros'. The 'Software' option is selected. At the bottom of the form is a blue button labeled 'Enviar'.

Haga clic en "Enviar" para enviar la incidencia.



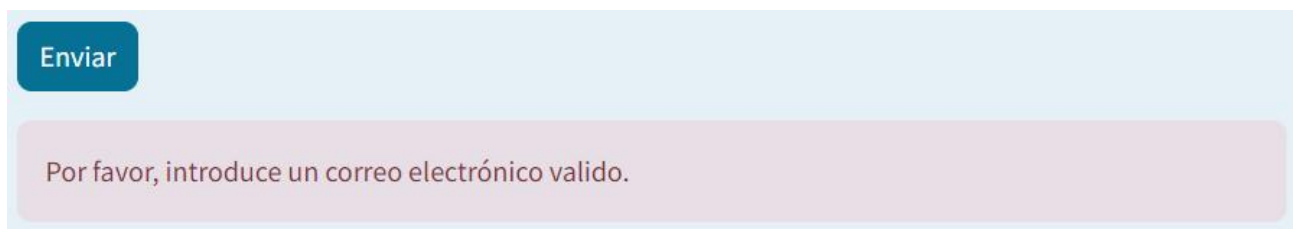
Respuestas posibles:

Campos vacíos: asegúrese de completar todos los campos.



The screenshot shows the form with a validation error. The 'Enviar' button is visible at the top left. Below it, a light pink error message box contains the text: 'Por favor, completa todos los campos.'

Correo electrónico no válido: ingrese un correo electrónico válido.



The screenshot shows the form with a validation error. The 'Enviar' button is visible at the top left. Below it, a light pink error message box contains the text: 'Por favor, introduce un correo electrónico valido.'

Envío realizado: confirmación de que la incidencia ha sido enviada.

Enviar

Disculpa las molestias. En breve nos pondremos en contacto contigo.

Apartado de Sugerencias:

[Información](#) [Soporte](#) [Sugerencias](#)

Sugerencias

Nos gustaría conocer tu opinión

Usuario

Email

Escribe tu sugerencia aquí:

☐ Deseo suscribirme a la versión Premium

Comentar

Complete el formulario con su sugerencia.

Puede optar por suscribirse a la versión premium utilizando el botón deslizable.

☐ Deseo suscribirme a la versión Premium

Haga clic en "Comentar" para enviar su sugerencia.

Comentar

Respuestas posibles:

Campos vacíos: asegúrese de completar todos los campos.

Comentar

Por favor, completa todos los campos.

Correo electrónico no válido: ingrese un correo electrónico válido si se suscribe.

☒ Deseo suscribirme a la versión Premium

Por favor, Introduzca su correo electrónico.

Envío realizado: confirmación de que la sugerencia ha sido enviada.

☐ Deseo suscribirme a la versión Premium

[Comentar](#)

¡Gracias por tu sugerencia!

Suscripción realizada: confirmación de suscripción a la versión premium con efectos visuales (globos de fiesta).



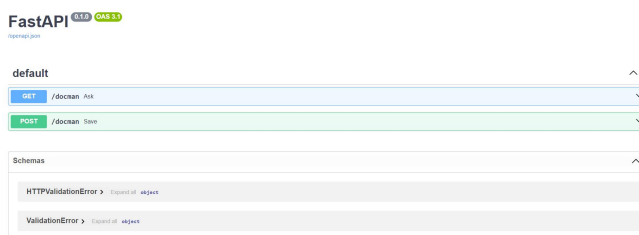
6. Pantalla API

Acceso a la API:



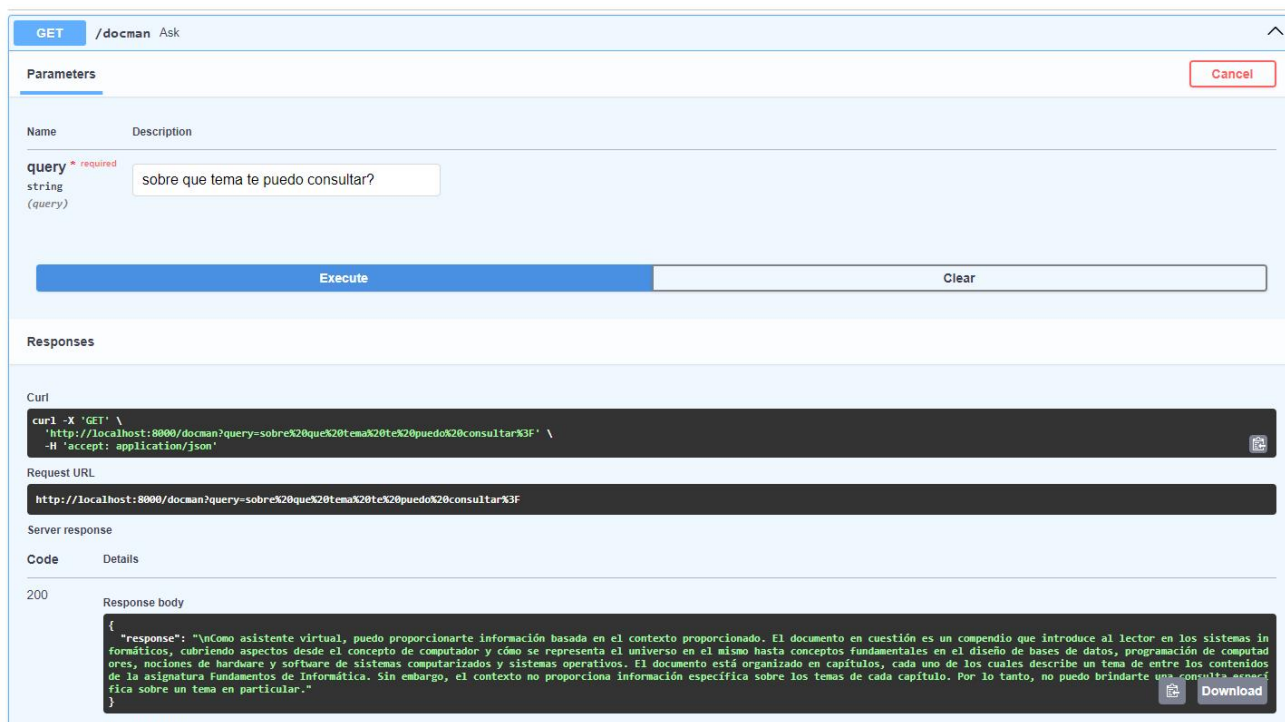
Haga clic en el enlace "API" en la barra de navegación.

Será dirigido a la documentación de la API desplegada en FastAPI.



Endpoints:

GET: Permite realizar consultas a la base de datos.



GET /docman Ask

Parameters

Name	Description
query * required string (query)	sobre que tema te puedo consultar?

Execute **Clear**

Responses

Code **Details**

200

Response body

```
{
  "response": "¡Como asistente virtual, puedo proporcionarte información basada en el contexto proporcionado. El documento en cuestión es un compendio que introduce al lector en los sistemas informáticos, cubriendo aspectos desde el concepto de computador y cómo se representa el universo en el mismo hasta conceptos fundamentales en el diseño de bases de datos, programación de computadores, nociones de hardware y software de sistemas computarizados y sistemas operativos. El documento está organizado en capítulos, cada uno de los cuales describe un tema de entre los contenidos de la asignatura Fundamentos de Informática. Sin embargo, el contexto no proporciona información específica sobre los temas de cada capítulo. Por lo tanto, no puedo brindarte una consulta específica sobre un tema en particular."
}
```

Download

POST: Permite almacenar documentación en la base de datos.



POST /docman Save

Parameters

Name	Description
data * required string (query)	data

Try it out

7. Enlace a Google

Realizar Búsquedas Externas:



Haga clic en el enlace "Google" en la barra de navegación.

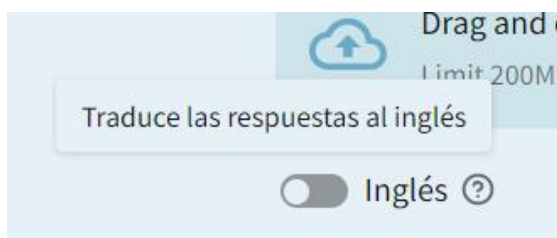
Será dirigido a la página principal de Google para realizar búsquedas externas.

8. Etiquetas de Ayuda

Visualización de Ayuda:

Para obtener ayuda sobre cualquier funcionalidad, pase el ratón sobre el signo de interrogación o sobre cualquier botón de la aplicación para ver la etiqueta correspondiente.

Pase el ratón sobre los interrogantes para ver etiquetas de ayuda.



Las etiquetas explican la funcionalidad del botón correspondiente.



7.2 Manual de instalación

Prerrequisitos

Antes de comenzar la instalación, asegúrese de tener los siguientes componentes instalados en su sistema:

- **Docker:** Asegúrese de tener Docker instalado y configurado en su máquina. Puede descargarlo desde Docker's official website.

- **Git:** Si desea clonar el repositorio desde un sistema de control de versiones, asegúrese de tener Git instalado. Puede descargarlo desde Git's official website.

Paso 1: Clonar el Repositorio

Si está utilizando un sistema de control de versiones, como Git, puede clonar el repositorio del proyecto:

```
git clone <URL_del_repositorio>cd <nombre_del_repositorio>
```

Reemplace <URL_del_repositorio> con la URL real del repositorio y <nombre_del_repositorio> con el nombre del directorio del proyecto.

Paso 2: Configuración del Archivo Docker Compose

Asegúrese de que el archivo docker-compose.yml esté presente en el directorio raíz del proyecto. Este archivo debe contener la configuración necesaria para construir y ejecutar los contenedores Docker.

Paso 3: Ejecutar los Contenedores Docker

Ejecute el siguiente comando para construir y ejecutar los contenedores Docker definidos en su archivo docker-compose.yml:

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

Este comando descargará las imágenes necesarias, construirá los contenedores y los ejecutará.

Paso 4: Acceder a la Aplicación

Una vez que los contenedores estén en funcionamiento, podrá acceder a la aplicación en su navegador web. Abra su navegador e ingrese la siguiente URL: <http://localhost:8501>

Paso 5: Documentación de la API

Si desea acceder a la documentación de la API, puede hacerlo visitando la siguiente URL:

<http://localhost:8000/docs>

8 Conclusiones y posibles ampliaciones

Conclusiones

- **Evaluación:** la aplicación no solo se distingue por su innovación tecnológica, sino también por su capacidad para mejorar la eficiencia operativa, la seguridad de los datos y la experiencia del usuario, consolidándose como una solución integral e innovadora para el manejo avanzado de información en tiempo real.
- **Dificultades:** Al utilizar el modelo LLM mediante su API de HuggingFaces, en ocasiones está sobrecargado de consultas simultaneas. Hay que esperar un tiempo y recargar el navegador para volver a realizar de nuevo la consulta.

Posibles Ampliaciones

- Soporte Multiplataforma: Extender compatibilidad a otros sistemas operativos.
- Gestión de usuarios.
- Añadir más tipos de documentos.
- Mejorar la interfaz de usuario.
- Almacenaje de las consultas en una base de datos para tener un histórico de consultas.

9 Bibliografía

- **Curso Udemy:** De cero a nivel profesional: aprende las claves de IA y construye las aplicaciones de IA Generativa con mayor potencial. 37 Horas.
<https://www.udemy.com/course/bootcamp-ia-generativa-y-aplicaciones-llm/?couponCode=ST2MT43024>
- **Video:** MEJORES y BARATOS: Cómo es que RAG está revolucionando los modelos de lenguaje.
<https://www.youtube.com/watch?v=P2m1kyGjAbA>
- **LlamaIndex:**
https://docs.llamaindex.ai/en/stable/getting_started/starter_example/

- **RAG:**

https://python.langchain.com/docs/use_cases/question_answering/#rag-architecture

<https://www.consultor365.com/ia/explicacion-rag/>

- **Modelos Fundacionales:**

<https://huggingface.co/models>

<https://mistral.ai/technology/#models>

<https://www.maginate.com/article/hands-on-with-new-mistral-mixture-of-experts/>

- **Machine Learning**

<https://xgboost.readthedocs.io/en/stable/tutorials/index.html>

- **Fastapi**

<https://fastapi.tiangolo.com/>

- **HTTPException** en Python.

<https://docs.python.org/es/3.9/library/http.client.html>

- **Ejemplos:**

<https://github.com/fferegrino/mananeras-rag/blob/main/rag-executed.ipynb>

- **Ejemplos:**

<https://github.com/fferegrino/llm-prompting/blob/main/api-explorer.ipynb>

10 Anexos

Toda la documentación y código correspondiente a este proyecto se encuentra en la siguiente URL de GitHub: github.com/CarmenGDAM/TFG