

GUÍA DE USUARIO

TABLA DE CONTENIDOS

I. FAST API Y ENDPOINTS

1. Introducción
2. Descripción General de las Tecnologías Utilizadas
3. Paso a paso de los Endpoints
4. Resumen de la Estructura del Código
5. Beneficios para el Negocio
6. Conclusión

II. SISTEMAS DE AGENTES DE ASISTENCIA

1. Introducción
2. Paso a Paso del Funcionamiento General
3. Descripción de Cada Agente
4. Resumen de la Estructura del Código
5. Beneficios para el Negocio
6. Conclusión

III. SISTEMAS RAG (RETRIEVAL AUGMENTE GENERATION)

1. Resumen Ejecutivo
2. Funcionalidades Principales
3. Implementaciones Disponibles
4. Requisitos de Implementación
5. Estructura del Código
6. Beneficios para el Negocio
7. Limitaciones y Consideraciones

FAST API Y ENDPOINTS

Documento de Resumen y Funcionamiento

1. Introducción

Esta sección explica el funcionamiento de la API desarrollada con FastAPI y la integración con un modelo de lenguaje de OpenAI, junto con la conexión a una base de datos MySQL alojada en Amazon Web Services (AWS). El objetivo es mostrar, de forma sencilla y paso a paso, cómo se gestionan las interacciones y se obtienen respuestas a través de diferentes endpoints, de modo que cualquier usuario, incluso sin conocimientos técnicos avanzados, pueda entender el proceso.

2. Descripción General de las Tecnologías Utilizadas

- **FastAPI**
 - Permite crear una API para interactuar remotamente con la aplicación.
 - Facilita la creación y gestión de endpoints (puntos de acceso) para diferentes funcionalidades.
 - **OpenAI (LLM: Large Language Model)**
 - Se utiliza un modelo de lenguaje (por ejemplo, GPT-4) para generar respuestas inteligentes basadas en preguntas o instrucciones.
 - **Base de Datos MySQL en AWS**
 - Se registra cada interacción y consulta, permitiendo mantener un historial de las operaciones realizadas por los alumnos y profesores.
-

3. Paso a Paso de los Endpoints

3.1. Página de Bienvenida ("/")

- **Qué hace:**
 - Al visitar la raíz de la API, se muestra un mensaje de bienvenida.
- **Propósito:**
 - Confirmar que la API está funcionando correctamente y lista para recibir solicitudes.

3.2. Prueba de Conexión a la Base de Datos ("/test-db")

- **Qué hace:**
 - Conecta a la base de datos MySQL alojada en AWS y ejecuta una consulta sencilla.

- **Propósito:**
 - Verificar que la conexión con la base de datos funciona correctamente

3.3. Inicio de Sesión para Profesores ("/login_prof")

- **Qué hace:**
 - Recibe el correo electrónico del profesor.
 - Valida sus credenciales en la base de datos.
 - Crea una nueva "sesión" para registrar la interacción.
 - Devuelve sus datos (nombre, email y bootcamp).
- **Propósito:**
 - Verificar a los profesores y proporcionar acceso a funcionalidades específicas (Dashboard).

3.4. Inicio de Sesión para Alumnos ("/login")

- **Qué hace:**
 - Recibe el correo del alumno.
 - Verifica su existencia en la base de datos.
 - Crea una nueva sesión y registra la hora de acceso
 - Devuelve los datos del alumno (nombre, vertical y área "Career").
- **Propósito:**
 - Identificar al alumno y preparar su sesión para interacciones posteriores.

3.5. Perfil del Alumno ("/perfil/{email}")

- **Qué hace:**
 - Busca al alumno por su correo electrónico.
 - Muestra su perfil, incluyendo módulos y unidades de su vertical (ej: Data Science) y de "Career Readiness".
- **Propósito:**
 - Proporcionar una vista completa del progreso y contenidos disponibles para el alumno.

3.6. Saludo del Chatbot ("/chatbot/greeting")

- **Qué hace:**
 - Devuelve un mensaje de saludo predeterminado.
- **Propósito:**
 - Establecer el tono de la interacción y ofrecer una bienvenida personalizada.

3.7. Selección de Vertical ("/selecciona")

- **Qué hace:**
 - Consulta la vertical (área de estudio) del alumno (ej: Data Science).
 - Presenta dos opciones: la vertical específica del alumno (por ejemplo, Data Science o Full Stack) y la opción "Career".

- **Propósito:**
 - Guiar al alumno para elegir entre contenidos técnicos o preparación profesional.

3.8. Registro de Elección ("/opción")

- **Qué hace:**
 - Recibe la opción seleccionada y el identificador de la sesión.
 - Verifica que la opción es válida y que la sesión corresponde al usuario.
 - Actualiza el registro de la sesión con la opción elegida.
- **Propósito:**
 - Personalizar las siguientes interacciones según la elección del alumno.

3.9. Listado de Módulos Disponibles ("/modulos/{vertical}")

- **Qué hace:**
 - Muestra los módulos disponibles para una vertical específica (ej: "Machine Learning" en Data Science).
 - Incluye identificadores únicos para cada módulo.
- **Propósito:**
 - Permitir al alumno explorar los módulos disponibles de su área de estudio.

3.10. Listado de Unidades de un Módulo ("/unidades/{modulo}")

- **Qué hace:**
 - Devuelve las unidades (temas) de un módulo específico (ej: "Pandas Básico" en Análisis de Datos).
 - Incluye identificadores únicos para cada unidad.
- **Propósito:**
 - Ayudar al alumno a navegar por contenidos detallados de un módulo.

3.11. Selección de Tema ("/temas/{id_unidad}/seleccionar")

- **Qué hace:**
 - Permite al alumno seleccionar un tema específico dentro de una unidad.
 - Almacena la selección en la base de datos.
- **Propósito:**
 - Guardar preferencias para personalizar futuras respuestas o recursos.

3.12. Generar Resumen de una Unidad ("/resumen/{vertical}")

- **Qué hace:**
 - Recibe identificadores de módulo y unidad, y verifica que corresponden a la vertical correcta.

- Utilizada OpenAI para generar un resumen técnico adaptado a la vertical (ej: explicación de Pandas para Data Science)
- Devuelve el resumen generado.
- **Propósito:**
 - Proporcionar un resumen comprensible y contextualizado para ayudar al alumno a entender mejor el tema.

3.13. Selección de Ayuda ("/selecciona_ayuda")

- **Qué hace:**
 - Ofrece opciones de ayuda según la vertical del alumno.
 - Ejemplo: "Career Readiness" muestra ayuda para CV o entrevistas.
- **Propósito:**
 - Brindar recursos específicos para resolver dudas o necesidades del alumno.

3.14. Registra de Ayuda ("/ayuda")

- **Qué hace:**
 - Recibe el tipo de ayuda selecciona (ej: "Preparación para entrevistas").
 - Actualiza la sesión con esta información.
 - **Propósito:**
 - Registrar la elección para proporcionar asistencia personalizada en siguientes interacciones.
-

4. Resumen de la Estructura del Código

1. Importaciones y Configuración Inicial

- **Librerías utilizadas:**
 - FastAPI: Para definir la API y sus endpoints.
 - Pydantic: Para definir modelos de datos y validar entradas.
 - pymysql: Para conectarse a la base de datos MySQL alojada en AWS.
 - OpenAI: Para interactuar con el modelo de lenguaje (GPT-4).
 - Logging: Para registrar eventos y depurar errores.
 - Unicorn: Para ejecutar el servidor de la API.
- **Configuración:**
 - Se define la clave de API para OpenAI (api_key).
 - Se inicializa el cliente de OpenAI (client).
 - Se configura el logging para seguimiento de errores.

2. Conexión a la Base de Datos

- **Función `get_db()`:**
 - Se encarga de establecer la conexión con la base de datos MySQL en AWS utilizando `pymysql`.
 - Configura los parámetros de conexión (host, usuario, contraseña, base de datos, puerto) y utiliza un cursor que devuelve diccionarios.

3. Definición de Modelos Pydantic

- **Modelos de datos para autenticación y sesión:**
 - **LoginRequest:** Valida el correo electrónico del usuario.
 - **LoginResponse:** Devuelve mensaje de éxito, ID de sesión y datos del alumno.
 - **LoginResponseProf:** Similar, pero para profesores (incluye bootcamp).
- **Modelos de interacción con el usuario:**
 - **SeleccionRequest:** Maneja la elección de opción (ej: módulo o career).
 - **ResumenRequest:** Define los parámetros para generar resúmenes (ID de módulo y unidad).
 - **UserQueryRequest:** Gestiona consultas del usuario según su vertical
- **Modelos de respuesta estructurada:**
 - **ModulosResponse, UnidadesResponse, TemasResponse:** Listan módulos, unidades o temas disponibles
 - **PerfilResponse:** Contiene datos completos del alumno, incluyendo módulos y unidades de su vertical y career.

4. Endpoints Principales

- **Endpoints básicos:**
 - **/ (GET):** Valida el correo del profesor, verifica su bootcamp asignado y devuelve sus datos.
 - **/test-db (GET):** Prueba la conexión a la base de datos mediante una consulta simple (`SELECT 1`).
- **Endpoints de verificación y perfil:**
 - **/login_prof (POST):** Valida el correo del profesor, verifica su bootcamp asignado y devuelve sus datos.
 - **/login (POST):** Crea una sesión para el alumno, registra su vertical y career, y devuelve un ID de sesión.
 - **/perfil/{email} (GET):** Muestra el perfil completo del alumno: datos personales, módulos y unidades de su vertical y career.
- **Endpoints de interacción con el Chatbot:**
 - **/chatbot/greeting (GET):** Devuelve un mensaje de saludo para iniciar la interacción con el usuario.
 - **/selecciona (POST):** Presenta al alumno las opciones de su vertical específica y la opción "Career Readiness".
 - **/opcion (POST):** Registra la opción elegida por el alumno (ej: un módulo) y actualiza la sesión en la base de datos.

- **Endpoints de consulta de módulos y unidades:**
 - **/modulos/{vertical} (GET):** Devuelve la lista de módulos disponibles para la vertical específica (ej: “Machine Learning” en Data Science).
 - **/unidades/{modulo} (GET):** Devuelve la lista de unidades (temas) para el módulo seleccionado.
 - **/temas/{id_unidad}/seleccionar (POST):**
 - Valida y registra la selección de un tema específico dentro de una unidad.
- **Endpoint para generar resumen de una unidad:**
 - **/resumen/{vertical} (POST):**
 - Recibe los identificadores de módulo y unidad, valida que la vertical sea correcta y construye un prompt.
 - Utiliza el modelo de OpenAI para generar y devolver un resumen del contenido.
- **Endpoints de ayuda y soporte:**
 - **/selecciona_ayuda (POST):** Ofrece opciones de ayuda según la vertical del usuario (ej: “Preparación de CV” para Career).
 - **/ayuda (POST):** Registra el tipo de ayuda seleccionada y actualiza la sesión.

5. Función Auxiliar y Utilidades

- **obtener_modulos_y_unidades():**
 - Función que ejecuta consultas para recuperar módulos de una vertical y, para cada módulo, sus unidades asociadas.
 - Organiza la información en una lista estructurada que se utiliza en la respuesta del endpoint /perfil/{email}.
 - **get_chat_response():**
 - Define prompts especializados según la vertical (Data Science, Full Stack, Career Readiness) y utiliza la API de OpenAI (GPT-4) para generar respuestas contextuales o consejos prácticos según la consulta del usuario.
-

5. Beneficios para el Negocio

- **Automatización y Escalabilidad:**
 - La API automatiza la interacción remota con el sistema educativo, permitiendo asistencia personalizada y escalable para múltiples usuarios.
- **Mejora en la Experiencia del Usuario:**
 - Las respuestas contextuales y estructuradas facilitan el acceso a la información y apoyan el proceso de aprendizaje.
- **Integración y Seguimiento:**
 - El registro de interacciones en la base de datos permite analizar el comportamiento de los usuarios y optimizar la oferta educativa.

- **Eficiencia e Historial:**
 - Permite obtener respuestas rápidas y contextualizadas sin depender exclusivamente de la intervención humana, además de mantener un historial para análisis y mejora continua.
- **Interacción Personalizada:**
 - Las respuestas se adaptan según la vertical de estudio y la elección del usuario, mejorando la experiencia de aprendizaje.

6. Conclusión

Esta parte integra tecnología: FastAPI, OpenAI, MySQL y AWS/GCP para ofrecer una experiencia interactiva en la que los alumnos pueden obtener información y orientación sobre distintos módulos y temas. Cada endpoint está diseñado para:

- Validar y consolidar la información del alumno.
- Registrar cada interacción de forma segura.
- Responder de manera contextual y estructurada.

La estructura y el proceso detallado aseguran que la información se maneje correctamente, facilitando tanto el uso como el análisis posterior de los datos, y aportando beneficios claros para el negocio y la experiencia educativa.

SISTEMA DE AGENTES DE ASISTENCIA

Documento de Resumen y Funcionamiento

1. Introducción

Esta sección está diseñada para apoyar a los estudiantes de The Bridge, ofreciéndoles asistencia en diversas áreas como investigación, resúmenes, soporte de estudio, práctica y orientación profesional. Cada "agente" es un componente especializado que se encarga de una tarea específica y, en conjunto, todos colaboran para responder preguntas y brindar apoyo a lo largo del proceso de aprendizaje.

2. Paso a Paso del Funcionamiento General

2.1. Objetivo General

- **Propósito:**
Ayudar a los estudiantes a obtener información, aclarar dudas, practicar contenidos y prepararse para el mundo profesional de manera interactiva y personalizada.

2.2. Descripción del Proceso

1. **Recepción de la Consulta del Estudiante**
 - **Qué hace:**
El sistema recibe la pregunta o consulta del estudiante.
 - **Propósito:**
Determinar cuál es la necesidad específica del alumno.
2. **Coordinación con el Agente Adecuado (StudyOrchestrator)**
 - **Qué hace:**
El coordinador analiza la consulta utilizando palabras clave y decide a qué agente redirigirla.
 - **Propósito:**
Asegurar que cada pregunta sea atendida por el componente especializado que pueda brindar la respuesta más adecuada.
3. **Redirección a un Agente Especializado**
 - **Agentes Disponibles:**
 - **ResearchAgent:** Para investigación y búsqueda de información.
 - **SummarizeAgent:** Para resumir textos extensos.
 - **StudySupportAgent:** Para explicar conceptos complejos.
 - **PracticeAgent:** Para generar ejercicios y evaluaciones.
 - **JobPrepAgent:** Para ofrecer orientación profesional.

- **Propósito:**
Cada agente se encarga de una tarea específica, garantizando respuestas precisas y contextualizadas.
 - 4. **Generación de la Respuesta**
 - **Qué hace:**
El agente seleccionado procesa la consulta (por ejemplo, realizando búsquedas, resúmenes o explicaciones) y envía el resultado de vuelta.
 - **Propósito:**
Proveer una respuesta completa y útil que apoye el proceso de aprendizaje del estudiante.
 - 5. **Entrega de la Respuesta**
 - **Qué hace:**
El sistema envía la respuesta al estudiante.
 - **Propósito:**
Facilitar al alumno el acceso a información, aclaraciones o ejercicios de práctica de manera inmediata.
-

3. Descripción de Cada Agente

3.1. Agente de Investigación (ResearchAgent)

- **Qué hace:**
Investiga información sobre un tema utilizando dos fuentes principales:
 - **Google:** Utiliza SerpAPI para buscar información en internet.
 - **W3Schools:** Extrae contenido del sitio educativo.
- **Cómo lo hace:**
Combina los resultados de ambas búsquedas y estructura una pregunta que se envía a un modelo de lenguaje (por ejemplo, GPT-4) para obtener una respuesta completa.
- **Propósito:**
Ofrecer a los estudiantes un panorama amplio y profundo de un tema antes de abordar preguntas específicas.

3.2. Agente de Resumen (SummarizeAgent)

- **Qué hace:**
Resume textos largos en un párrafo breve y conciso.
- **Cómo lo hace:**
Envía el contenido a un modelo de lenguaje que genera un resumen resaltando los puntos y conceptos clave.
- **Propósito:**
Facilitar la comprensión de temas extensos y ayudar a los estudiantes a captar rápidamente lo esencial.

3.3. Agente de Soporte de Estudio (StudySupportAgent)

- **Qué hace:**
Proporciona explicaciones detalladas de conceptos complejos.
- **Cómo lo hace:**
Responde a preguntas específicas explicando paso a paso el tema, identificando errores comunes y ofreciendo consejos.
- **Propósito:**
Ayudar a los estudiantes a comprender de forma clara temas que pueden parecer difíciles o confusos.

3.4. Agente de Práctica (PracticeAgent)

- **Qué hace:**
Genera ejercicios prácticos, pruebas de conocimientos y autoevaluaciones.
- **Cómo lo hace:**
Basado en un tema dado, crea:
 - Un ejercicio práctico con instrucciones y solución de ejemplo.
 - Un conjunto de preguntas de opción múltiple para evaluar la comprensión.
 - Una autoevaluación con conceptos clave y recomendaciones.
- **Propósito:**
Permitir a los estudiantes practicar y evaluar su conocimiento de manera interactiva y estructurada.

3.5. Agente de Preparación Profesional (JobPrepAgent)

- **Qué hace:**
Ofrece consejos y orientaciones para la preparación laboral, tales como la creación de currículum, optimización de perfiles de LinkedIn y preparación para entrevistas.
- **Cómo lo hace:**
Responde a consultas relacionadas con la carrera profesional proporcionando pasos concretos y estrategias.
- **Propósito:**
Ayudar a los estudiantes a prepararse para el mundo laboral, desarrollando habilidades prácticas para buscar empleo y destacar en entrevistas.

3.6. Coordinador de Estudio (StudyOrchestrator)

- **Qué hace:**
Actúa como director del sistema, decidiendo a cuál de los agentes enviar la consulta del estudiante.
- **Cómo lo hace:**
Analiza la pregunta y, mediante la identificación de palabras clave, redirige la consulta al agente más adecuado.

- **Propósito:**
Garantizar que cada pregunta se maneje por el componente especializado que ofrezca la respuesta óptima.
-

4. Resumen de la Estructura del Código

4.1. Imports y Configuración

- **Qué hace:**
Se importan las librerías necesarias para conectarse a OpenAI, realizar búsquedas en Google, parsear HTML, hacer solicitudes HTTP y cargar variables de entorno.
- **Propósito:**
Configurar el entorno para que los agentes puedan funcionar correctamente.

4.2. Agentes y sus Métodos

1. **ResearchAgent**
 - **Objetivo:** Investigar información a partir de una consulta.
 - **Métodos:**
 - `google_search(query)`: Realiza una búsqueda en Google usando SerpAPI.
 - `get_w3schools_content(query)`: Extrae información de W3Schools.
 - `research(query)`: Combina ambas fuentes para construir un prompt y obtener una respuesta detallada del modelo de lenguaje.
2. **SummarizeAgent**
 - **Objetivo:** Resumir contenidos extensos.
 - **Método:**
 - `summarize(content)`: Toma un texto y devuelve un resumen breve con los puntos clave.
3. **StudySupportAgent**
 - **Objetivo:** Proporcionar explicaciones detalladas de conceptos complejos.
 - **Método:**
 - `explain(query)`: Explica paso a paso un concepto o tema solicitado.
4. **PracticeAgent**
 - **Objetivo:** Crear ejercicios, pruebas y autoevaluaciones.
 - **Método:**
 - `create_exercise(topic)`: Genera ejercicios prácticos, preguntas de opción múltiple y una autoevaluación estructurada.

5. **JobPrepAgent**
 - **Objetivo:** Brindar orientación profesional para la búsqueda de empleo.
 - **Método:**
 - `provide_guidance(query)`: Ofrece consejos para mejorar currículum, perfil de LinkedIn y preparación para entrevistas.
 6. **StudyOrchestrator**
 - **Objetivo:** Coordinar la consulta del estudiante y redirigirla al agente adecuado.
 - **Método:**
 - `handle_request(query)`: Analiza la consulta y dirige la pregunta al agente especializado correspondiente.
 7. **Otros Componentes Complementarios**
 - **PineconeRetriever:** Realiza búsquedas en índices vectoriales para recuperar documentos relevantes.
 - **CVAnalyzer:** Analiza currículums en formato PDF, identificando áreas de mejora a través de la generación y consulta de embeddings.
-

5. Beneficios para el Negocio

- **Personalización y Especialización:**

Cada agente se especializa en una función, ofreciendo respuestas precisas y adaptadas a las necesidades específicas de los estudiantes.
 - **Mejora del Aprendizaje:**

La combinación de investigación, resúmenes, explicaciones y ejercicios facilita la comprensión de temas complejos y refuerza el aprendizaje.
 - **Automatización del Soporte Educativo:**

La automatización permite una asistencia continua, reduciendo la necesidad de intervención manual y escalando el soporte a más usuarios.
 - **Optimización y Análisis:**

La integración de servicios en la nube y tecnologías de IA permite registrar y analizar interacciones para mejorar continuamente la experiencia educativa.
 - **Preparación Profesional:**

Los agentes de orientación laboral ayudan a los estudiantes a prepararse eficazmente para el mundo profesional, fortaleciendo habilidades prácticas para la búsqueda de empleo.
-

6. Conclusión

El sistema de agentes de asistencia para bootcamp integra tecnología de punta (modelos de lenguaje avanzados, búsquedas automatizadas y procesamiento de datos) para ofrecer una experiencia interactiva y personalizada a los estudiantes. Cada agente, desde la investigación hasta la preparación profesional, está diseñado para:

- Atender necesidades específicas del aprendizaje.
- Proporcionar respuestas completas y contextualizadas.
- Facilitar la interacción y el progreso en el proceso educativo.

Esta estructura no solo mejora la experiencia de los alumnos, sino que también aporta beneficios claros al negocio, al automatizar el soporte educativo y optimizar el seguimiento y análisis de cada interacción.

Sistema RAG (Retrieval Augmented Generation)

Documento de Resumen y Funcionamiento

1. Resumen Ejecutivo

- **Definición:**
 - Una herramienta inteligente que permite buscar y consultar información en diferentes tipos de documentos (notebooks de Data Science, notebooks de Full Stack, archivos PDF).
 - Funciona de manera natural, como si se conversara con un profesor experto.
 - **Objetivo:**
 - Implementar un sistema de Generación Aumentada por Recuperación (RAG) que combine IA y búsqueda vectorial.
 - Integrar información de proyectos Full Stack y datasets de Kaggle para generar insights y análisis detallados.
 - **Beneficios:**
 - Ofrece respuestas contextuales y precisas mediante la recuperación de documentos relevantes.
 - Facilita la toma de decisiones y la obtención de análisis detallados.
 - **Tecnología Utilizada:**
 - Inteligencia Artificial para entender preguntas.
 - Recuperación de información mediante técnicas de búsqueda en documentos.
-

2. Funcionalidades Principales

- **Indexación de Contenido:**
 - Procesa documentos automáticamente según su tipo.
 - Divide el contenido en fragmentos manejables.
 - Crea una base de datos localizable con la información indexada.
 - Mantiene índices separados para cada tipo de documento.
- **Sistema de Consulta:**
 - Permite hacer preguntas en lenguaje natural.
 - Proporciona respuestas contextualizadas mediante múltiples consultas.
 - Adapta las respuestas según el dominio (Data Science, Full Stack, etc.).
- **Personalización:**
 - Adaptable a distintos tipos de documentos.
 - Ajuste de parámetros como el tamaño de los fragmentos.

- Configurable para diversas necesidades de búsqueda, con índices específicos por dominio.
-

3. Implementaciones Disponibles

1. **RAG para Notebooks de Data Science (rag_kaggle.ipynb)**
 - Especializado en contenido de ciencia de datos.
 - Indexa notebooks con ejemplos prácticos y tutoriales.
 - Ideal para consultas sobre análisis de datos, machine learning y estadística.
 2. **RAG para Notebooks de Full Stack (rag_fs.ipynb)**
 - Enfocado en desarrollo web full stack.
 - Cubre temas de frontend, backend y herramientas de desarrollo.
 - Perfecto para consultas sobre programación y mejores prácticas.
 3. **RAG para Documentos PDF (rag_pdf.ipynb)**
 - Procesa documentos PDF de cualquier temática.
 - Permite indexar manuales, documentación técnica y papers.
 - Versátil para diferentes tipos de contenido textual.
-

4. Requisitos de Implementación

- **Entorno Python:**
 - Configurado con librerías como Langchain, OpenAI, MistralAI, Pinecone, BeautifulSoup, entre otras.
 - **Archivo .env:**
 - Almacena claves sensibles (ej.: PINECONE_API_KEY, MISTRAL_API_KEY).
 - **Servicios en la Nube:**
 - Acceso a servicios como Pinecone para gestionar índices vectoriales.
 - **Modelos de Lenguaje Avanzados:**
 - Acceso a modelos como ChatOpenAI y ChatMistralAI para respuestas basadas en prompts estructurados.
 - **Datos de Entrada:**
 - Procesamiento de documentos (PDFs, notebooks) dividiéndolos en fragmentos y generando embeddings.
-

5. Estructura del Código

1. **Carga de Variables de Entorno:**
 - Uso de la librería dotenv para cargar claves API y configuraciones sensibles.
2. **Inicialización de Servicios Externos:**

- **Pinecone:** Inicializado con la API key para gestionar índices vectoriales.
 - **Clientes de Modelos de Lenguaje:** Configurados para MistralAI y OpenAI.
 - **Herramientas de Langchain:** Utilizadas para estructurar prompts y dividir documentos en fragmentos.
3. **Proceso de Generación Aumentada por Recuperación (RAG):**
- **Carga y División de Documentos:**
 - Uso de herramientas como PyPDFLoader y RecursiveCharacterTextSplitter para dividir el contenido en fragmentos.
 - **Generación de Embeddings y Almacenamiento:**
 - Transformación de cada fragmento en un embedding con el modelo MistralAI y almacenamiento en un índice de Pinecone.
 - **Búsqueda de Información:**
 - Realización de consultas en el índice de Pinecone para recuperar los fragmentos más relevantes.
 - **Generación de Respuestas:**
 - Construcción de un prompt con el contexto recuperado y generación de respuesta con un modelo de lenguaje (por ejemplo, ChatMistralAI).
 - **Limpieza de Recursos:**
 - Eliminación de archivos temporales y limpieza del índice para optimizar recursos.
-

6. Beneficios para el Negocio

- **Automatización y Eficiencia:**
 - Automatiza el proceso de recuperación de información y generación de respuestas.
 - Reduce el tiempo de análisis y facilita la toma de decisiones.
 - **Mejora en la Toma de Decisiones:**
 - Permite obtener insights detallados a partir de proyectos Full Stack y datasets de Kaggle.
 - **Escalabilidad:**
 - Utiliza servicios en la nube y modelos avanzados que permiten ampliar la solución según la demanda.
 - **Valor Estratégico:**
 - Combina datos estructurados y no estructurados para ofrecer respuestas contextualizadas y de alto valor añadido.
-

7. Limitaciones y Consideraciones

- **Dependencias:**
 - Requiere conexión a Internet y claves API válidas.
 - Consume recursos según el volumen de consultas y necesita espacio de almacenamiento.
- **Costes:**
 - Capas gratuitas disponibles; uso intensivo puede requerir planes premium.
 - Considerar límites de uso mensual.
- **Precisión:**
 - Depende de la calidad de los documentos indexados.
 - Puede requerir validación humana en casos críticos y ajustes específicos según el tipo de documento.