

ID	HU1
Tarea	Implementación básica de interfaz de chatbot
Tipo	feature
HU	Yo como usuario quiero interactuar con una interfaz similar a las usadas por los chatbots actuales como ChatGPT, Deepseek, etc.
Criterio aceptación	- La interfaz permite al usuario escribir un mensaje y enviarlo con un botón o con la tecla Enter. - Se muestran los mensajes del usuario y las respuestas del chatbot en formato de conversación. - La interfaz tiene un diseño visual limpio y sencillo, similar al de chatbots modernos.

ID	HU2
Tarea	Crear el contexto que da identidad al asistente
Tipo	feature
HU	Yo como desarrollador, quiero configurar un prompt base que guíe al chatbot para que se enfoque en temas de Ingeniería de Sistemas, para su comportamiento futuro.
Criterio aceptación	- El código del chatbot incluye un mensaje de sistema o prompt base explícito que instruye al modelo a enfocarse en temas de Ingeniería de Sistemas. - El prompt base está escrito en un lenguaje claro y específico, e incluye indicaciones sobre rol, alcance temático y tono de respuesta.

ID	HU3
Tarea	Implementar ChromaDB y el modelo de embeddings de OpenAI
Tipo	feature
HU	Yo como desarrollador, quiero que el chatbot obtenga sus respuestas a partir de los resultados de búsqueda por similitud realizados en ChromaDB, utilizando fragmentos de documentos procesados como embeddings mediante el modelo de OpenAI, para garantizar que las respuestas estén con el contexto del contenido cargado.
Criterio aceptación	- Los documentos pdfs son procesados y divididos en fragmentos (chunks) para ser vectorizados. - Cada fragmento es convertido en un embedding utilizando el modelo de OpenAI y almacenado correctamente en ChromaDB.

	Al recibir una pregunta del usuario, el sistema consulta ChromaDB y recupera los fragmentos usando búsqueda por similitud.
--	--

ID	HU4
Tarea	Crear archivo .gitignore
Tipo	feature
HU	Yo como desarrollador, quiero que la aplicación cuente con un archivo .gitignore configurado correctamente, para evitar que se suban al repositorio archivos temporales, de prueba o en constante cambio, para mantener un control limpio del código.
Criterio aceptación	El archivo .gitignore está presente en el repositorio y contiene reglas que excluyen correctamente archivos y carpetas temporales, de prueba o en constante cambio (__pycache__/, .env, *.log, *.pyc, db/, etc.), evitando que se suban al control de versiones.

ID	HU5
Tarea	Mejorar el prompt de la variable de contexto del chatbot
Tipo	refactor
HU	Yo como desarrollador, quiero que el contexto proporcionado al chatbot esté claramente estructurado y definido, para asegurar que el modelo lo interprete correctamente y genere respuestas coherentes.
Criterio aceptación	El mensaje de sistema del chatbot está redactado de forma clara, estructurada y alineada con el propósito del proyecto, y se valida que al ser incluido se generan respuestas dentro del dominio establecido.

ID	HU6
Tarea	Procesar el prompt del usuario para mejorar la búsqueda
Tipo	feature
HU	Yo como desarrollador, quiero que el prompt del usuario sea procesado mediante técnicas como la lematización o expandir la consulta con términos semejantes.
Criterio aceptación	La consulta del usuario es lematizada correctamente, eliminando palabras vacías (stopwords), y generando una versión simplificada y generalizada del texto original.

	• La consulta es ampliada con sinónimos y términos relacionados generados mediante un llamado a GPT. • Las técnicas de lematización y expansión semántica son integradas correctamente en el flujo de procesamiento antes de ejecutar la búsqueda de información.
--	--

ID	HU7
Tarea	Implementar búsqueda híbrida con BM25
Tipo	feature
HU	Yo como desarrollador, quiero que el chatbot obtenga información a través de dos enfoques de búsqueda, para reducir posibles sesgos y mejorar la precisión en la recuperación de información.
Criterio aceptación	• El sistema implementa dos enfoques distintos de búsqueda: uno basado en similitud semántica mediante embeddings y otro basado en búsqueda tradicional (BM25). • Ambos enfoques son ejecutados al momento de procesar una consulta del usuario, y sus resultados pueden combinarse o evaluarse para mejorar la relevancia de las respuestas.

ID	HU8
Tarea	Implementar el procesamiento de pdfs a texto plano
Tipo	feature
HU	Yo como desarrollador, quiero que los pdfs sean procesados a texto plano antes de generar los fragmentos que serán utilizados para crear la base de datos de embeddings para facilitar su lectura.
Criterio aceptación	• Los archivos pdf son convertidos exitosamente a texto plano sin pérdida de contenido. • El texto extraído conserva la estructura lógica de los párrafos, eliminando saltos de línea innecesarios o errores de segmentación. • El texto plano resultante es utilizado como base para generar los fragmentos que luego serán convertidos en embeddings y almacenados en la base de datos.

ID	HU9
Tarea	Implementar una funcionalidad para resumir los txt

Tipo	feature
HU	Yo como desarrollador, quiero que los archivos de texto plano sean resumidos y guardados para tener información más concisa y condensada para ocupar menos ventana de contexto en las llamadas al modelo de GPT.
Criterio aceptación	- Los archivos de texto plano son procesados correctamente y su contenido es resumido sin perder las ideas principales del texto original. - Los resúmenes generados contienen información condensada, clara y relevante, reduciendo significativamente la longitud respecto al texto original.

ID	HU10
Tarea	Implementar el guardado en cache las conexiones a ChromaDB y embeddings.
Tipo	feature
HU	Yo como desarrollador, quiero que se carguen en cache las conexiones a ChromaDB y el modelo de embeddings de OpenAI para mejorar el rendimiento y cargar de forma más rápida los datos que se buscan en la base de datos.
Criterio aceptación	- La conexión a ChromaDB se almacena en caché utilizando @st.cache_resource, evitando que se cree una nueva instancia en cada ejecución. - La carga del modelo de embeddings de OpenAI también se almacena en caché, garantizando que solo se inicialice una vez por sesión. - No se generan errores de recarga ni conflictos por el uso de recursos compartidos en las funciones almacenadas en caché.

ID	HU11
Tarea	Implementar la función de OCR
Tipo	feature
HU	Yo como desarrollador, quiero que la aplicación pueda leer contenido de pdfs escaneados mediante funcionalidad OCR utilizando Tesseract y PyTesseract, para garantizar la extracción de texto de documentos que no contienen datos digitales.
Criterio aceptación	La aplicación identifica correctamente cuándo un PDF contiene imágenes en lugar de texto digital y activa el

	procesamiento OCR de forma automática o controlada. • Se utiliza Tesseract OCR, a través de PyTesseract, para extraer texto desde imágenes contenidas en los pdfs escaneados. • El texto extraído desde los pdfs escaneados se guarda correctamente como texto plano y puede ser tratado igual que los pdfs con texto digital.
--	--

ID	HU12
Tarea	Crear archivo requirements.txt
Tipo	feature
HU	Yo como desarrollador, quiero que el proyecto cuente con un archivo requirements.txt correctamente configurado, para facilitar la instalación de todas las dependencias necesarias en cualquier entorno de desarrollo.
Criterio aceptación	• El archivo requirements.txt está presente en la raíz del proyecto y contiene todas las librerías necesarias para su ejecución. • Todas las versiones de las dependencias están especificadas de forma compatible para evitar conflictos durante la instalación. • Al ejecutar pip install -r requirements.txt en un entorno limpio, el proyecto se instala correctamente sin errores de dependencias faltantes. • El archivo no incluye librerías innecesarias o propias del entorno local que no sean requeridas por el proyecto.

ID	HU13
Tarea	Eliminar la funcionalidad de resumir textos
Tipo	refactor
HU	Yo como desarrollador, quiero eliminar la función de resumen de textos del procesamiento, para preservar la integridad del contenido original y evitar la pérdida de información relevante durante la recuperación.
Criterio aceptación	• La función de resumen de textos ha sido completamente eliminada del flujo de procesamiento de documentos sin afectar el funcionamiento de la aplicación.

ID	HU14
Tarea	Refactorizar la búsqueda híbrida de información.
Tipo	refactor
HU	Yo como desarrollador, quiero que la aplicación implemente correctamente los mecanismos de recuperación de información mediante búsqueda por similitud con embeddings y mediante BM25 de forma independiente, para asegurar flexibilidad y precisión en el proceso de consulta.
Criterio aceptación	• La búsqueda por similitud utilizando embeddings funciona de forma independiente de la búsqueda basada en BM25. • La lematización se elimina del flujo de procesamiento para los textos que se utilizan en la creación de embeddings, con el fin de conservar el contexto completo del contenido original. • Los fragmentos utilizados para ambas estrategias de búsqueda se almacenan en un archivo .pkl, incluyendo la versión original de los fragmentos y la versión lematizada. • Los resultados de ambas búsquedas se combinan, manteniendo la prioridad según su puntuación o relevancia. • Los fragmentos duplicados se eliminan en el proceso de combinación para evitar redundancia en el contexto entregado al modelo.

ID	HU15
Tarea	Estandarizar el nombre de los archivos pdfs
Tipo	enhancemet
HU	Yo como desarrollador, quiero que los archivos pdf utilizados como fuente de información tengan nombres estandarizados, claros y descriptivos, para facilitar su identificación, organización y mantenimiento dentro del proyecto.
Criterio aceptación	• Los archivos pdf deben seguir una convención de nombres estandarizada con el siguiente formato: tipo_documento_numero_dia_mes_año como por ejemplo acuerdo_006_jun_28_2023.pdf. • El nombre del archivo debe ser claro y descriptivo, reflejando de manera precisa el contenido o tipo de documento por ejemplo: acuerdo, resolución, manual, informe, etc.

	• Todos los archivos deben evitar el uso de espacios, caracteres especiales o ambigüedades en el nombre, utilizando guiones bajos (_) para separar los elementos.
--	---

ID	HU16
Tarea	Crear el resto de las vistas de la aplicación
Tipo	feature
HU	Yo como usuario, quiero acceder a diferentes vistas dentro de la aplicación donde pueda consultar una descripción del contenido de los pdfs utilizados como fuente de información y visualizar los documentos embebidos, para comprender mejor el contexto de las respuestas del chatbot.
Criterio aceptación	• La aplicación debe incluir una barra lateral (sidebar) con navegación entre vistas utilizando st.page y st.navigation, permitiendo al usuario acceder fácilmente a cada sección informativa. • Cada vista debe mostrar una descripción clara y concisa del contenido del pdf correspondiente, destacando su propósito, tema principal y posibles aportes al sistema. • Los documentos pdf deben estar embebidos directamente en la interfaz, permitiendo su lectura sin necesidad de descargarlos o abrirlos externamente. • Los pdf embebidos deben ser completamente responsive, adaptándose correctamente a diferentes tamaños de pantalla. • Las vistas deben estar organizadas de forma clara e intuitiva, con títulos representativos y sin ambigüedades sobre el contenido del documento visualizado. • Se debe mover la lógica del asistente a una vista aparte, usando como enrutador el archivo main.

ID	HU17
Tarea	Reorganizar la estructura del proyecto
Tipo	refactor
HU	Yo como desarrollador, quiero organizar la estructura del proyecto distribuyendo los archivos en carpetas según su función, para facilitar el mantenimiento, la escalabilidad y la comprensión del código.

Criterio aceptación	- Yo como desarrollador, quiero organizar la estructura del proyecto distribuyendo los archivos en carpetas según su función, para facilitar el mantenimiento, la escalabilidad y la comprensión del código. - Debe existir una carpeta assets/ que contenga todos los elementos gráficos utilizados en la interfaz de la aplicación (íconos, logos, imágenes, etc.). - Debe existir una carpeta data/ que contenga dos subcarpetas: pdfs/ donde se almacenan los documentos fuente, texts/ donde se guardan los archivos de texto plano generados a partir de los pdfs. - Debe existir una carpeta src/ que contenga: core/ con los archivos de lógica principal de la aplicación (por ejemplo: procesamiento de consulta, búsqueda, carga de datos), util/ con funciones auxiliares o de utilidad general (por ejemplo: limpieza de texto, configuración de variables, carga de modelos, etc.), views/ donde se ubican las distintas vistas de la aplicación (por ejemplo: página principal, descripción de pdfs, visualización del chatbot, etc.). - En el directorio raíz del proyecto deben encontrarse: El archivo main.py como punto de entrada principal de la aplicación. El archivo requirements.txt con todas las dependencias necesarias correctamente listadas. El archivo bm25_corpus.pkl con los datos de soporte para la búsqueda basada en BM25. La carpeta db/ que contiene la base de datos ChromaDB local utilizada para la búsqueda por embeddings.
---------------------	--

ID	HU18
Tarea	Cambiar el renderizado de la respuesta del asistente.
Tipo	enhancement
HU	Yo como usuario, quiero que las respuestas del asistente se muestren progresivamente a medida que se generan, para recibir retroalimentación inmediata y mejorar la experiencia de interacción.
Criterio aceptación	- La respuesta del asistente debe mostrarse token por token a medida que se genera, sin esperar a que se complete todo el mensaje. - Debe incluirse un elemento visual (como un spinner, puntos animados, o mensaje de "Escribiendo...") que indique que el asistente está generando una respuesta.

ID	HU19
Tarea	Crear el archivo readme
Tipo	feature
HU	Yo como desarrollador, quiero que el repositorio del proyecto cuente con un archivo README.md bien estructurado, que describa claramente los pasos necesarios para instalar, configurar y ejecutar la aplicación, con el fin de facilitar su uso y despliegue por otros desarrolladores o usuarios técnicos.
Criterio aceptación	• El repositorio debe contener un archivo README.md ubicado en la raíz del proyecto. • El archivo README.md debe incluir al menos las siguientes secciones: Descripción general del proyecto, instrucciones de instalación y cómo ejecutar la aplicación. • Toda la información del README.md debe estar disponible en español e inglés.