

Propuesta 1: MCP Server “KB Local”

Descripción

El servidor MCP KB Local permite indexar y consultar documentos locales (PDF, Markdown y TXT) para que el chatbot pueda responder preguntas con fragmentos de texto citados desde los archivos reales.

Funcionalidades principales (tools):

- `index_files(dir)`: Indexa todos los archivos soportados en el directorio indicado.
- `search(query, top_k)`: Devuelve los pasajes más relevantes junto con el nombre del archivo y posición.
- `open(path, page_range | byte_range)`: Abre un fragmento específico del documento.

Especificación técnica

- Formato de entrada/salida: JSON-RPC conforme a MCP.
- Restricciones: Lista blanca de directorios y límite de tamaño de archivo.
- Dependencias: Python, pypdf, markdown-it-py, rapidfuzz o FAISS para búsquedas.

Ejemplo de uso en el chatbot

1. Usuario: “Busca información sobre Dijkstra en mis apuntes.”
 - Chatbot llama a `search("Dijkstra")` → devuelve fragmentos de `apuntes.pdf` y `notas.md`.
2. Usuario: “Muéstrame completo el segundo pasaje.”
 - Chatbot llama a `open("apuntes.pdf", page_range="5-6")`.
3. El chatbot puede luego guardar un resumen en un repo usando Git MCP.

Propuesta 2: MCP Server “Genius Notes”

Descripción

El servidor MCP Genius Notes procesa notas de clase o proyectos escritas en Markdown para generar resúmenes, tarjetas de estudio (flashcards) y listas de tareas extraídas automáticamente.

Funcionalidades principales (tools):

- `summarize_note(path, style="bullet|narrative")`: Genera un resumen breve o narrativo de una nota.
- `generate_flashcards(path, n)`: Devuelve pares de pregunta-respuesta listos para estudiar.
- `extract_todos(dir)`: Escanea notas en un directorio y lista todos los TODOs encontrados.

Especificación técnica

- Formato de entrada/salida: JSON-RPC conforme a MCP.
- Restricciones: Solo procesa archivos .md.
- Dependencias: Python, markdown-it-py, reglas basadas en regex.

Ejemplo de uso en el chatbot

1. Usuario: "Resúmeme la nota de Sistemas Operativos."
 - Chatbot llama a `summarize_note("so_clase1.md", "bullet")`.
2. Usuario: "Genera 10 flashcards para repasar antes del quiz."
 - Chatbot llama a `generate_flashcards("so_clase1.md", 10)`.
3. Usuario: "Muéstrame todos mis pendientes de las notas."
 - Chatbot llama a `extract_todos("./notas/")`.