**Proyecto de Inteligencia artificial para dar clases de medicina (Lector de pdf con ChatBot)**

Nombre: Felipe Diaz Aimar

**Primera iteración:**

**✅ Requerimientos**

**Funcionales:**

1. El sistema debe poder **Mostrar contenido de la catedra** (infografías o videos y artículos relevantes.)
2. El sistema debe permitir **Descargar contenido de la catedra** contenido médico o de la catedra.
3. El sistema debe poder **Mostrar exámenes de autoevaluación.**
4. El sistema debe permitir al usuario **Cargar apuntes** con contenido médico o de la catedra.
5. El sistema debe poder **Mostrar apunte cargado** mientras interactúa con el chatbot.
6. El sistema debe poder **Mostrar otro apunte** **cargado**.
7. El sistema debe poder **Registrar una pregunta** mediante un asistente virtual (chatbot).
8. El sistema debe poder **Mostrar la respuesta del asistente.**

**No funcionales:**

1. El sistema debe funcionar en navegadores modernos (Chrome, Firefox, Edge).
2. El sistema debe ser **accesible** desde computadoras y dispositivos móviles.
3. El procesamiento de datos debe ser **rápido y seguro**.
4. El chatbot debe responder en **lenguaje natural y comprensible**.
5. El chatbot debe respetar **límites de contexto** para evitar respuestas fuera de tema.
6. Se debe priorizar el uso de **tecnología gratuita u open source** (por ejemplo, modelos gratuitos en OpenRouter).
7. El sistema debe tener una **interfaz web amigable**.

**Requisitos técnicos:**

* Soporte para archivos .pdf.
* Framework web: HTML, CSS, JavaScript.
* Backend opcional: Node.js / Python (Flask o FastAPI si hace falta procesar los PDFs).
* Uso de APIs como OpenAI, OpenRouter u otra que permita embeddings o LLMs.

**🔍 Análisis**

**1. Análisis del problema**

Actualmente, los estudiantes de medicina deben estudiar apuntes extensos, muchas veces en formato PDF, que contienen terminología compleja y fragmentos densos de información. El problema principal es la **dificultad para encontrar información puntual** **o entenderla** dentro de los textos y la **falta de una guía interactiva** que les ayude a comprender los temas en tiempo real.

Por eso, se busca desarrollar una solución basada en inteligencia artificial que permita:

* Visualizar apuntes en PDF.
* Descargar apuntes en PDF.
* Cargar apuntes en PDF.
* Consultar esos apuntes mediante preguntas.
* Recibir respuestas claras, precisas y contextuales.

**2. Usuarios y actores involucrados**

* **Estudiantes de medicina:** usuarios finales, consultan los apuntes mediante el chatbot.
* **Docentes:** pueden validar las respuestas y usar el sistema como apoyo didáctico.
* **Equipo de desarrollo:** diseñadores, programadores y testers.
* **Sistema de IA externo (LLM):** modelo que genera las respuestas basadas en los textos cargados.
* **Plataforma de ejecución (navegador):** donde el usuario interactúa con el sistema.

**3. Requisitos informativos**

* El sistema debe poder acceder al contenido completo del PDF.
* Es necesario implementar algún método para **buscar o recuperar información de modo de tener un historial en la conversación**.
* Se debe garantizar que el chatbot **no invente información**.

**4. Requisitos de procesamiento**

* Extracción del texto desde PDFs (incluso con imágenes, tablas o estructuras complejas).
* Envío del contenido relevante al modelo de lenguaje con el prompt del usuario.

**5. Análisis de viabilidad**

* **Técnica:** existen herramientas para la extracción de texto (pdf.js, pdfplumber) y APIs accesibles para IA (OpenRouter, Hugging Face).
* **Económica:** si se usa infraestructura gratuita, no hay costos elevados. Ideal para universidades o proyectos académicos.
* **Temporal:** puede desarrollarse en etapas. La versión mínima viable (MVP) podría estar lista en pocas semanas.
* **Legal/Ética:** se debe verificar que los contenidos de los PDFs no infrinjan derechos de autor si se usa en entornos públicos o compartidos.

**6.Casos de Uso: ARCHIVO ADJUNTO “Diagrama Casos de uso.mdj”**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

**6.Descripcion de casos de uso:**

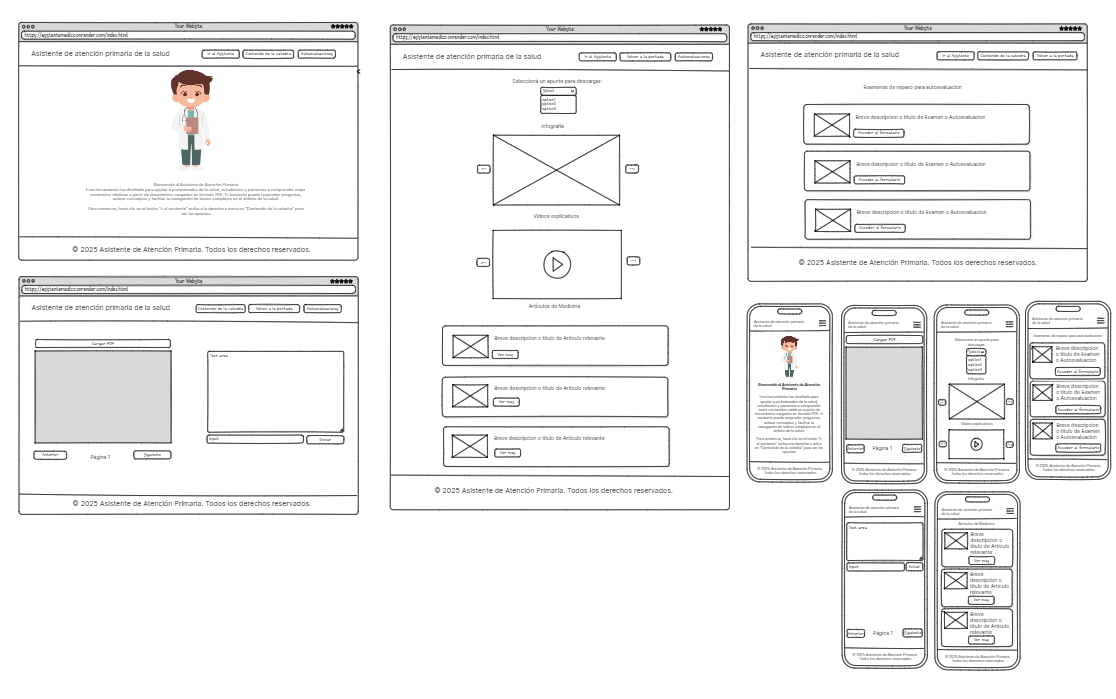
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caso de Uso**: Registrar pregunta | | | | |
| **ID: 1** | | | **Fecha:** | |
| **Descripción**: Registrar una pregunta y enviarla | | | | |
| **Actores Principales**: Persona | | **Actores Secundarios**: Asistente virtual | | |
| **Observaciones:** | | | | |
| **Precondiciones**: - | | | | |
| **Post- Condiciones** | **Éxito:** Se registra una nueva pregunta con estado “Answered” o “Error” | | | |
| **Fracaso:** No seregistra una nueva pregunta con estado “Answered” o “Error” | | | |
| Flujo Principal | | | | Flujo Alternativo |
| 1. **El caso de uso inicia cuando** la **Persona** accede al sitio web para realizar una consulta. | | | |  |
| 2. La **Persona** carga un apunte para realizar una pregunta al mismo. | | | |  |
| 3. La **Persona** escribe el prompt para realizar una **pregunta**. | | | |  |
| 4. Se verifica si el archivo está cargado correctamente y lo está. | | | | 4.1 Se verifica si el archivo está cargado correctamente y no lo está.  4.2 La persona escribe un nuevo prompt.  4.3 Se sigue con el paso 5. |
| 5. Se envía la pregunta a el **Asistente Virtual.** | | | |  |
| 6. La consulta es exitosa. | | | | 6.1 La consulta no es exitosa.  6.2 Se instancia una nueva pregunta en estado “Error" y se muestra en el historial de consultas.  6.3 Se sigue con el paso 9. |
| 7. Para gestionar la pregunta, incluir al caso de uso “Mostrar respuesta del Asistente” | | | |  |
| 8. Se instancia un nuevo pedido en estado “**Answered**” y se muestra en el historial de consultas. | | | |  |
| 9. Finaliza el caso de uso. | | | |  |

**7.Diagramas Business Process Model and Notation (BPMN):**

ARCHIVOS ADJUNTOS: “DBPMN 1-8.BPMN”

**Diseño:**

ARCHIVO ADJUNTO “PROTOTIPO.F0”



**Implementación:**

<https://asistentemedico.onrender.com/>

**Testing:**

ARCHIVO ADJUNTO “TestCaseAV.XSLX”