Documentación para "Mi Aplicación de Lectura"

1. Resumen General

"Mi Aplicación de Lectura" es una herramienta educativa diseñada para mejorar las habilidades de lectura y pensamiento crítico de los estudiantes. Permite a los usuarios cargar textos, interactuar con una inteligencia artificial (IA) para analizar el contenido y generar notas de estudio personalizadas para el aprendizaje espaciado. La aplicación se centra en desarrollar la **literacidad crítica**, ayudando a los estudiantes a comprender, analizar y evaluar textos en contextos sociales, políticos y culturales.

- **Público objetivo**: Estudiantes de diferentes niveles educativos que buscan fortalecer sus habilidades de lectura y análisis crítico.

2. Funcionalidades Clave

La aplicación ofrece las siguientes funcionalidades integradas para proporcionar una experiencia de aprendizaje completa:

2.1. Carga de Texto

- **Descripción**: Los usuarios pueden cargar textos para leerlos y analizarlos en la aplicación.
- **Métodos de carga**:
  - Subir archivos en formato txt, pdf, docx.
  - Pegar texto directamente en un área de texto (textarea).
- **Requisitos funcionales**:
  - Validar que los archivos sean txt, pdf, docx y mostrar mensajes de error si no lo son.
  - Asegurarse de que el texto pegado o subido se procese correctamente.
  - Limpiar el área de carga tras subir el contenido para evitar confusiones.

2.2. Visor de Texto

- **Descripción**: El texto cargado se muestra en un visor cómodo y legible.
- **Funcionalidades**:
  - Permitir ajustes de tamaño de fuente y contraste para accesibilidad.
  - Posibilidad de resaltar o anotar el texto en futuras versiones.
- **Requisitos funcionales**:
  - Diseño responsive para adaptarse a diferentes dispositivos.
  - Interfaz limpia y sin distracciones.

2.3. Interacción con IA (Chat)

- **Descripción**: Un componente de chat donde la IA hace preguntas sobre el texto y facilita discusiones.

- **Tipos de preguntas**:
  - **Literal**: Información explícita (ej. "¿Qué dijo el personaje X?").
  - **Inferencial**: Requiere deducción (ej. "¿Por qué crees que X actuó así?").
  - **Crítico-valorativo**: Reflexión personal (ej. "¿Estás de acuerdo con la decisión de X?").
- **Discusión contextual**: La IA aborda el contexto social, político y cultural del texto para fomentar la literacidad crítica.
- **Requisitos funcionales**:
  - Generar preguntas dinámicas basadas en el texto cargado.
  - Permitir respuestas del usuario y ofrecer retroalimentación o nuevas preguntas.
  - Mantener una conversación fluida y relevante.

2.4. Control de Atención

- **Descripción**: Minimizar distracciones para mantener al usuario enfocado en la lectura y el análisis.
- **Funcionalidades**:
  - Deshabilitar la selección y copia de texto en el visor.
  - *(Opcional)* Modo pantalla completa o bloqueo de cambio de pestañas.
- **Requisitos funcionales**:
  - Garantizar que el usuario permanezca concentrado en la tarea.

2.5. Progreso y Evaluación

- **Descripción**: Guardar el progreso del usuario y ofrecer una evaluación final.
- **Funcionalidades**:
  - Almacenar las respuestas a las preguntas de la IA.
  - Generar una evaluación basada en la calidad, profundidad de las respuestas, sintaxis, ortografía.
- **Requisitos funcionales**:
  - Sistema de retroalimentación claro y motivador.
  - Guardado automático del progreso.

2.6. Notas de Estudio para Aprendizaje Espaciado - Función Avanzada

Notas de Estudio para Aprendizaje Espaciado - Función Avanzada

Descripción General
La aplicación utiliza IA para generar **notas de estudio personalizadas** a partir de textos cargados por el usuario. Estas notas están diseñadas para facilitar el **aprendizaje espaciado**, adaptándose automáticamente al tipo de texto (narrativo, poético, filosófico, ensayo, etc.) y ofreciendo un **cronograma inteligente de repaso** basado en los principios de la curva del olvido.

Características clave

1. **Extracción Inteligente de Contenido**
   - Clasificación automática del tipo de texto (o selección manual).
   - Análisis semántico para identificar el 20% más relevante del contenido.

    - Notas estructuradas y concisas, optimizadas para la retención.
2. **Notas Personalizadas por Tipo de Texto**
    - **Narrativo**: Personajes, espacio/tiempo, voz narrativa, tema.
    - **Poético**: Objeto poético, recursos literarios, métrica y rima.
    - **Filosófico**: Ideas fundamentales, preguntas clave, argumentos.
    - **Ensayo**: Tesis, ideas secundarias, evidencia, estructura lógica.
3. **Cronograma Dinámico de Aprendizaje Espaciado**
    - El usuario puede elegir la duración total del proceso (por ejemplo: 1 mes, 2 semanas, etc.) o seguir un esquema sugerido.
    - El sistema propone fechas de repaso automático según un patrón como este:

Repaso	Intervalo sugerido
1º	Día 1 (estudio inicial)
2º	Día 2
3º	Día 4
4º	Día 7
5º	Día 14
6º	Día 30 (opcional)

    - Las fechas se ajustan dinámicamente si el usuario se atrasa o adelanta.
4. **Recordatorios y Seguimiento**
    - El usuario recibe notificaciones automáticas para cada sesión de repaso.
    - Cada nota tiene un botón para marcar como “repasada”.
    - El sistema adapta el calendario si detecta olvido o falta de repaso (mediante mini tests opcionales o autoevaluación rápida).
5. **Panel de Control del Usuario**
    - Visualización del progreso de cada conjunto de notas.
    - Opciones para editar el intervalo de repaso manualmente.
    - Sugerencias automáticas de repaso reforzado si el usuario falla en recordar una nota.

3. Detalles Técnicos y Estructura

La aplicación se desarrolla con **React.js**, utilizando una arquitectura modular para facilitar el mantenimiento y la escalabilidad. Aquí están los detalles técnicos clave:

3.1. Componentes Modulares

- CargaTexto.js:
    - Maneja la subida de archivos .txt y el pegado de texto.
    - Envía el texto procesado al componente principal (App.js).
- VisorTexto.js:
    - Muestra el texto cargado en un formato legible.
    - Recibe el texto como prop desde App.js.
- ChatIA.js:
    - Gestiona el chat con la IA, incluyendo preguntas y respuestas.
    - Integra la lógica para interactuar con la API de IA.

- NotasEstudio.js:
    - Permite al usuario seleccionar el tipo de texto y la cantidad de notas.
    - Muestra las notas generadas por la IA y permite guardarlas.
- App.js:
    - Componente principal que coordina los demás, gestiona el estado global (texto cargado, respuestas, etc.) y pasa datos entre componentes.

3.2. Integración de la IA

- **API sugerida**: OpenAI (ChatGPT).
- **Configuración**:
    - Almacenar la clave de la API en un archivo .env (REACT_APP_OPENAI_API_KEY).
    - Usar la biblioteca openai para realizar solicitudes desde ChatIA.js y NotasEstudio.js.
- **Funcionalidad**:
    - Enviar el texto cargado a la API para generar preguntas y notas relevantes.
    - Adaptar las preguntas y notas según el nivel del usuario y el tipo de texto.

3.3. Accesibilidad

- **Funcionalidades**:
    - Texto a voz para usuarios con discapacidades visuales.
    - Ajustes de contraste y tamaño de fuente.
- **Requisitos técnicos**:
    - Usar bibliotecas como react-speech para texto a voz.
    - Implementar CSS dinámico para ajustes de estilo.

3.4. Control de Atención

- **Implementación**:
    - Deshabilitar selección de texto con CSS (user-select: none).
    - *(Opcional)* Detectar cambios de pestaña con eventos JS (visibilitychange, blur) y mostrar mensajes o pausar la sesión.

3.5. Almacenamiento de Progreso y Notas

- **Sugerencia**: Usar Firebase o una base de datos similar.
- **Funcionalidad**:
    - Guardar respuestas, progreso y notas de estudio.
    - Generar evaluaciones finales con la ayuda de la IA.

---

4. Flujo de Uso

Aquí está el flujo típico de la aplicación desde la perspectiva del usuario:

1. **Cargar Texto**:
    - El usuario sube un .txt o pega texto.
    - La aplicación valida y muestra el texto en el visor.
2. **Leer el Texto**:
    - El usuario lee el texto en un entorno libre de distracciones.
3. **Interactuar con la IA**:
    - La IA inicia con una pregunta literal.
    - El usuario responde en el chat.
    - La IA evalúa la respuesta y avanza a preguntas más complejas o inicia una discusión contextual.
    - El intercambio continúa hasta completar la sesión.
4. **Generar Notas de Estudio**:
    - El usuario selecciona el tipo de texto (si no es automático) y cuántas notas desea.
    - La IA genera notas basadas en el 20% más importante del contenido, adaptadas al tipo de texto.
    - El usuario revisa y guarda las notas para estudiar a largo plazo.
5. **Evaluación y Progreso**:
    - Las respuestas y notas se guardan automáticamente.
    - Al finalizar, se muestra una evaluación basada en el análisis del usuario.

---

5. Ejemplo Práctico de Notas de Estudio

Texto Narrativo

- **Entrada**: Fragmento de una novela.
- **Solicitud**: 5 notas.
- **Salida**:
    1. Personajes principales: Juan y María.
    2. Espacio: Ciudad ficticia en los años 80.
    3. Tiempo interno: Una semana de invierno.
    4. Voz narrativa: Tercera persona omnisciente.
    5. Tema: La lucha por la supervivencia.

Texto Poético

- **Entrada**: Un poema.
- **Solicitud**: 3 notas.
- **Salida**:
    1. Objeto poético: La luna como símbolo de soledad.
    2. Figuras literarias: Metáfora ("luna de cristal").
    3. Rima: Versos libres sin métrica fija.

---

6. Resumen para el Programador

"Mi Aplicación de Lectura" es una herramienta educativa que combina lectura guiada, interacción con IA y generación de notas de estudio para mejorar las habilidades de lectura y pensamiento crítico. Utiliza **React.js** con componentes modulares (CargaTexto.js, VisorTexto.js, ChatIA.js, NotasEstudio.js, App.js) y se integra con una API como **OpenAI** para generar preguntas y notas adaptadas al tipo de texto.

Tareas clave para implementar:

- Implementar la carga de texto y validación en CargaTexto.js.
- Desarrollar el visor en VisorTexto.js con ajustes de accesibilidad.
- Integrar la API de OpenAI en ChatIA.js para preguntas dinámicas.
- Crear NotasEstudio.js para generar notas basadas en el tipo de texto y el 20% más importante del contenido.
- Añadir control de atención (CSS/JS).
- Configurar almacenamiento y evaluación con Firebase u otra base de datos.