**Informe de Estado Actual y Hoja de Ruta Técnica**

**1. Estado Actual del Proyecto (Post-Reorganización)**

Hemos superado con éxito la fase más crítica de configuración y limpieza. El estado actual es:

* **✅ Repositorio Profesional:** La estructura de carpetas está limpia y organizada, separando frontend, backend, docs, scripts y tests.
* **✅ Backend Operativo:** El servidor de Python (Flask) arranca sin errores de dependencias tras la reconstrucción del requirements.txt y la solución de los problemas de compilación.
* **✅ Frontend Operativo:** El servidor de Next.js arranca, compila el código TypeScript y renderiza la estructura básica de los componentes sin errores.
* **✅ Herramientas de Desarrollo:** Disponemos de scripts .bat para automatizar el arranque y parada de los servidores, facilitando el flujo de trabajo.

**Conclusión del Estado Actual:** Tenemos una base técnica sólida, limpia y funcional. Ahora podemos dejar de "arreglar" y empezar a "construir".

**2. Hoja de Ruta Técnica para Alcanzar la Visión Final**

Esta hoja de ruta nos llevará desde nuestro estado actual hasta la aplicación descrita en tu docs\_resumen\_final\_anclora.docx.

**Fase 1: Conexión y Funcionalidad Core (Nuestro Foco Inmediato)**

* **Objetivo:** Hacer que la aplicación sea funcionalmente útil en su versión más básica.
* **Tareas:**
  1. **Conectar Lógica del Frontend al Backend:** Reemplazar todas las funciones simuladas en services/api.ts y AuthContext.tsx por llamadas fetch reales a los endpoints del backend que ya están definidos en main.py (/login, /register, /analyze-file, etc.).
  2. **Implementar Flujo de Conversión Completo:** Asegurar que un usuario pueda subir un archivo, que este se envíe al backend, se procese (inicialmente con una conversión simple) y se pueda descargar el resultado.
  3. **Activar el Sistema de Créditos:** Conectar la lógica del frontend para que las conversiones descuenten créditos del usuario en la base de datos a través de la API.
  4. **Puesta a Punto del Diseño:** Aplicar los últimos retoques de Tailwind CSS para que la interfaz coincida 100% con las maquetas del "Anclora Workspace".

**Fase 2: Profesionalización del Flujo de Trabajo y Pruebas**

* **Objetivo:** Asegurar la calidad y mantenibilidad del código a largo plazo.
* **Tareas:**
  1. **Configurar Linting y Formateo:** Añadir ESLint y Prettier al package.json del frontend para mantener un código consistente y limpio.
  2. **Implementar la Batería de Pruebas:** Crear y ejecutar las pruebas unitarias y de integración (con Vitest) para el frontend y el backend, utilizando la estructura que definimos en la carpeta tests/.
  3. **Establecer un Flujo de Git Profesional:** Formalizar el uso de ramas (feature/, fix/) para el desarrollo y requerir Pull Requests para fusionar cambios a la rama main.

**Fase 3: Desarrollo de Funcionalidades Avanzadas (La Visión)**

* **Objetivo:** Implementar los diferenciadores únicos descritos en tu resumen.
* **Tareas:**
  1. **Desarrollar el Motor de Optimización Inteligente:** Implementar en el backend la lógica para analizar rutas de conversión A→B→C vs. A→C .
  2. **Crear el Tema Dinámico:** Desarrollar el hook useTimeBasedTheme.tsx y la lógica para cambiar los colores de la aplicación según la hora del día .
  3. **Construir el Selector de Formatos "Abanico":** Diseñar el componente FormatSelector.tsx con las animaciones y la UX revolucionaria que definiste .
  4. **Implementar el Chatbot Inteligente:** Desarrollar el componente SmartChatbot.tsx y conectarlo a un servicio de IA para proporcionar soporte contextual .