CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

PROGRAMACIÓN PARA INTERNET

Sección: D03

Proyecto. Better | Essay

Profesor: Michel Emanuel López Franco

Alumna: Katia Marlene Salcedo Huerta

Código: 210588774

Contenido

Resumen

Better | Essay es una aplicación web que integra tecnologías de inteligencia artificial para ayudar a los usuarios en la mejora de sus textos, tanto ensayos como resúmenes.

Se usa el modelo GPT-4o-mini por medio de OpenRouter para la corrección y reescritura de ensayos, y DistilBART CNN-12-6 (Hugging Face Transformers) para la generación de resúmenes precisos, dentro de una arquitectura en la que se utilizan: React, Node.js, FastAPI y MongoDB Atlas.

El objetivo de la plataforma es ofrecer una herramienta práctica, accesible para perfeccionar la escritura académica y profesional.

Objetivo general

Desarrollar una aplicación web capaz de analizar, corregir y resumir ensayos y generar resúmenes claros mediante el uso de modelos de inteligencia artificial.

Objetivos específicos

- Implementar una interfaz para ingresar y visualizar ensayos.
- Implementar una interfaz para ingresar texto extenso y visualizar resúmenes.
- Manejar usuarios y autenticación JWT.
- Utilizar MongoDB Atlas para el almacenamiento de datos en la nube.
- Desplegar los servicios en Render y Hugging Face Spaces.

Arquitectura general del sistema

Componente	Tecnología	Función principal
Frontend	React + Vite + TypeScript	Interfaz de usuario, manejo de formularios y vistas.
Backend	Node.js + Express + TypeScript	API principal, autenticación, conexión base de datos.
Base de datos	MongoDB Atlas	Almacenamiento seguro de usuarios.
IA Corrección	OpenRouter (GPT-4o-mini)	Revisión de ensayos, corrección y sugerencias.
IA Resumen	FastAPI + Transformers (DistilBART CNN-12-6)	Generación de resúmenes de texto.
Despliegue	Render + Hugging Face Spaces	Infraestructura.

Explicación técnica de las inteligencias artificiales

Corrección de ensayos – OpenRouter (GPT-4o-mini)

Lenguaje: TypeScript

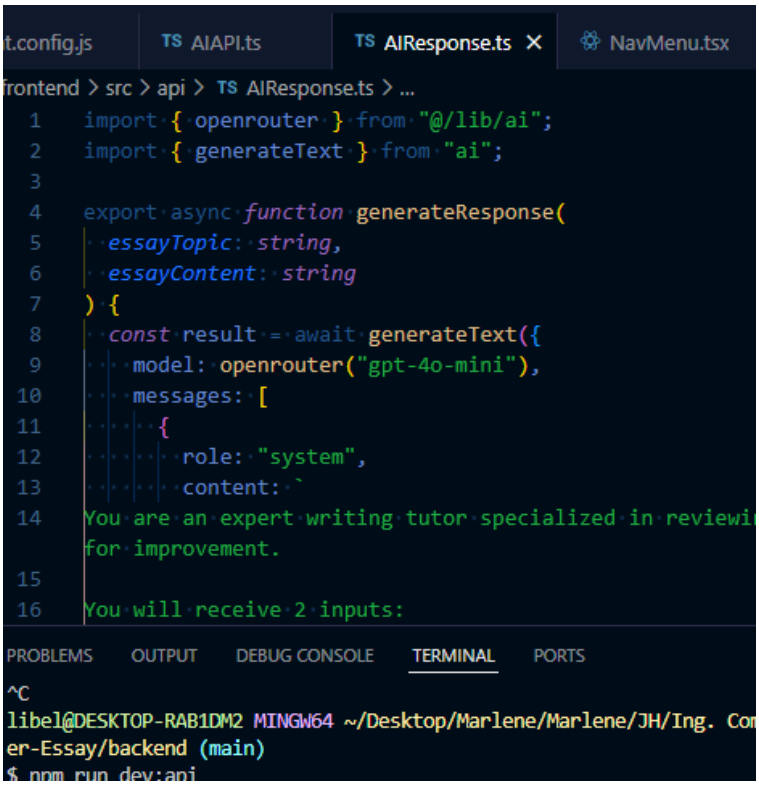
El servicio de corrección se basa en el modelo GPT-4o-mini, accedido mediante la API de OpenRouter.
El código envía al modelo dos entradas: el tema y el contenido del ensayo, junto con un mensaje del sistema que define el rol del modelo como un *tutor experto en escritura*.

El prompt está hecho para producir una salida estructurada en formato Markdown, divida en tres secciones:

1. **Correcciones y Feedback:** errores ortográficos, gramaticales y de estilo, con sugerencias específicas.

2. **Ensayo Mejorado:** una versión reescrita, clara y coherente, conservando las ideas del usuario.
3. **Recomendaciones Generales:** consejos para mejorar redacción, cohesión y vocabulario.

Fragmento clave del código:



Generación de resúmenes – FastAPI + DistilBART (Hugging Face)

Lenguaje: Python

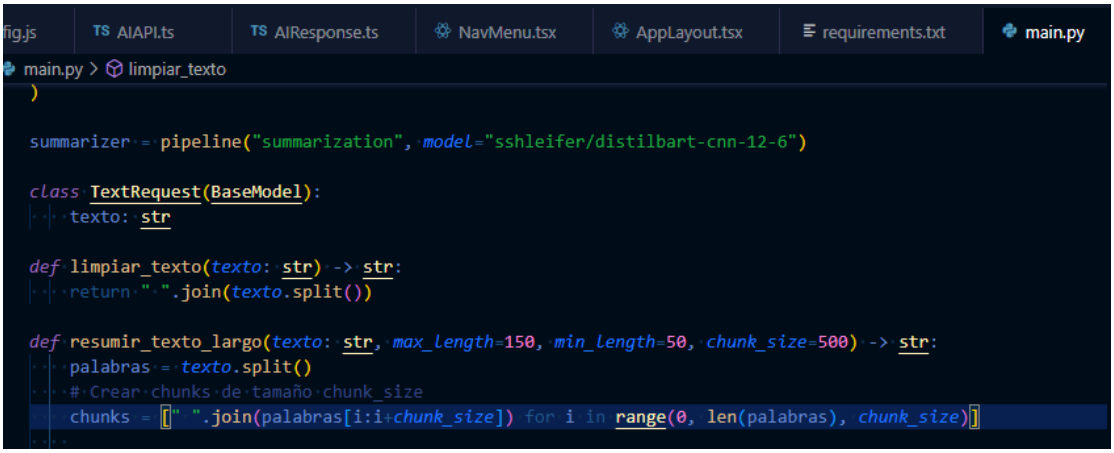
La API de resúmenes está desarrollada con FastAPI, y utiliza el modelo sshleifer/distilbart-cnn-12-6, una versión optimizada del modelo BART, para tareas de *abstractive summarization*.

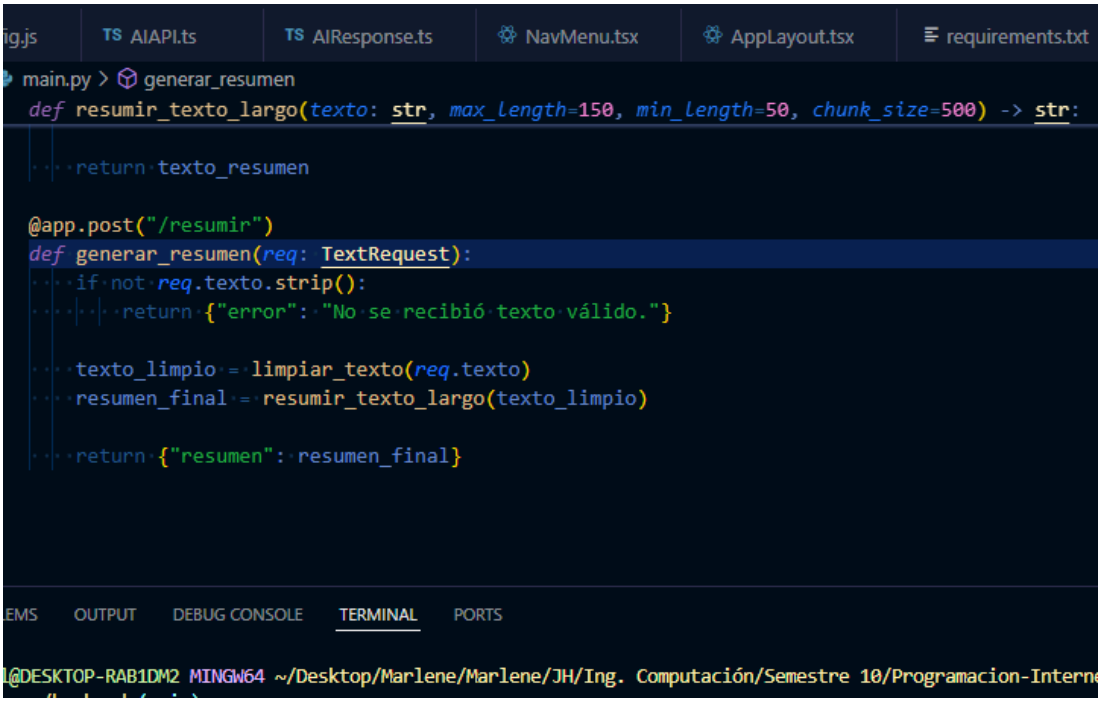
El sistema implementa una función llamada resumir_texto_largo() que divide el texto en fragmentos de hasta 500 palabras. Cada fragmento es procesado individualmente y luego los resultados parciales se combinan y refinan para generar un resumen final.

Flujo del proceso:

1. El usuario envía el texto desde el frontend al endpoint /resumir.
2. FastAPI limpia el texto y lo fragmenta en partes manejables.
3. Cada parte se pasa al modelo summarizer.
4. Se unen los resultados parciales y se ajusta la longitud del resumen final.

Fragmento clave del código:





Despliegue

- **Frontend y Backend:** Render
- **IA de Resúmenes:** Hugging Face Spaces
- **Base de Datos:** MongoDB Atlas
- **Corrección de Ensayos:** OpenRouter API

Se tienen variables de entorno (.env, .env.local) que protegen las claves privadas.