

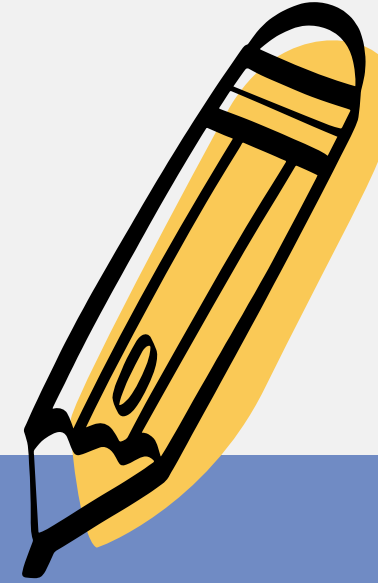
Programación para Internet

A solid yellow horizontal bar.

PROYECTO FINAL BETTER | ESSAY

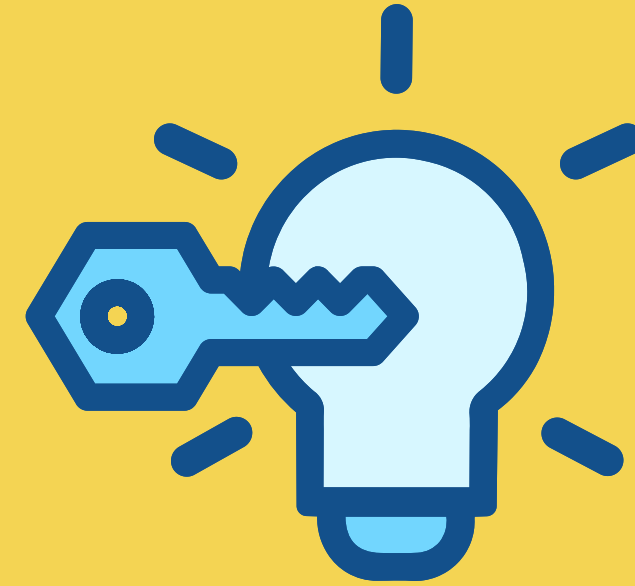
React · Node.js · MongoDB Atlas · FastAPI
· OpenRouter · Typescript

Katia Marlene Salcedo Huerta



Descripción

Better | Essay utiliza inteligencia artificial para corregir ensayos y generar resúmenes automáticos, ayudando a los usuarios a mejorar su escritura y comprensión textual.



OBJETIVO GENERAL

Desarrollar una app web que use IA para analizar, corregir y mejorar ensayos y además sirva para resumir textos de forma automática, clara y coherente.

React

Typescript

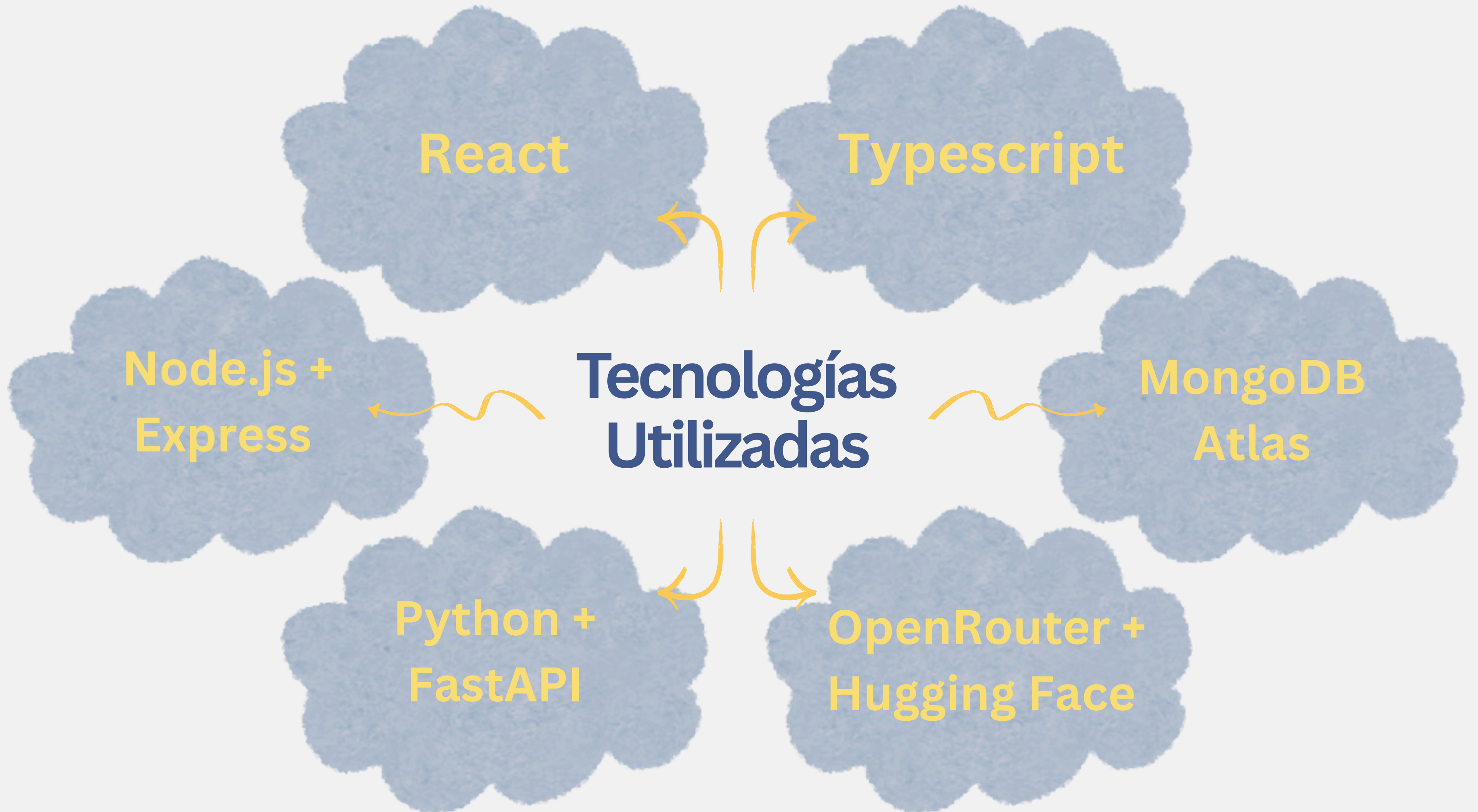
Tecnologías Utilizadas

Node.js +
Express

MongoDB
Atlas

Python +
FastAPI

OpenRouter +
Hugging Face



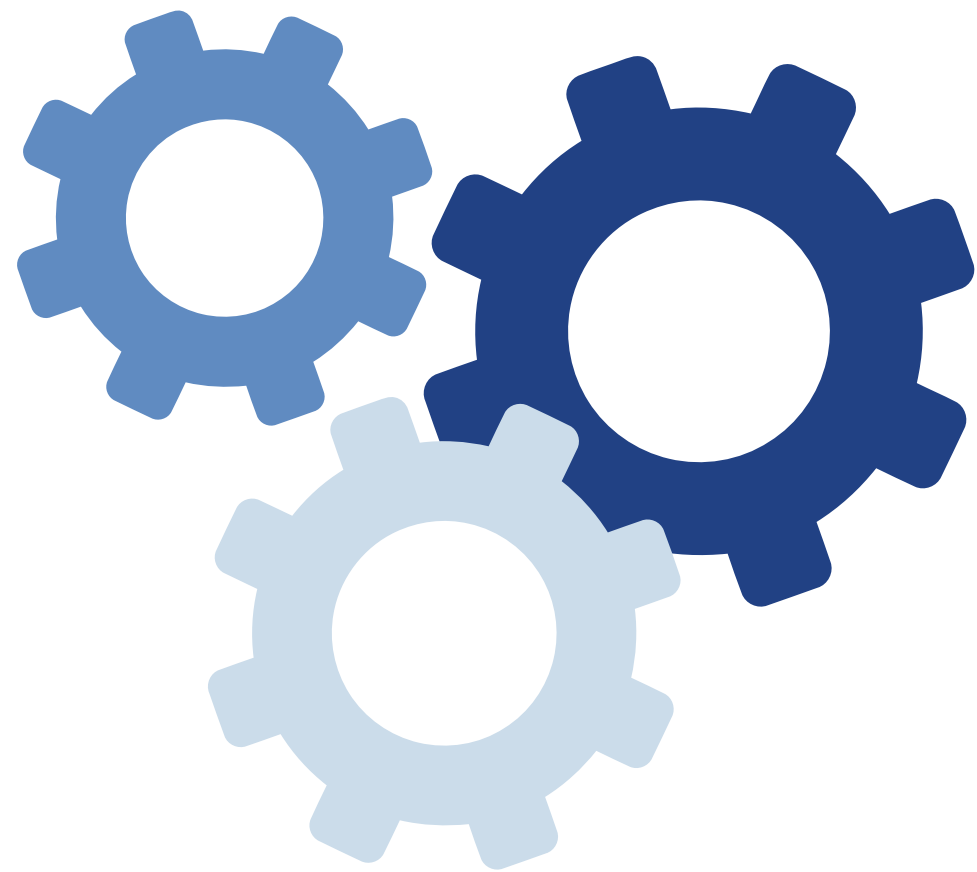
ARQUITECTURA Y PROGRAMACIÓN DE SISTEMAS

El sistema está estructurado en módulos independientes pero conectados, lo que permite:

- Escalabilidad
- Seguridad
- Eficiencia en el procesamiento



ARQUITECTURA Y PROGRAMACIÓN DE SISTEMAS



Se implementa una separación clara de responsabilidades mediante dos backends:

- Backend principal: gestión de usuarios y autenticación
- Backend de IA: procesamiento de texto y generación de resúmenes

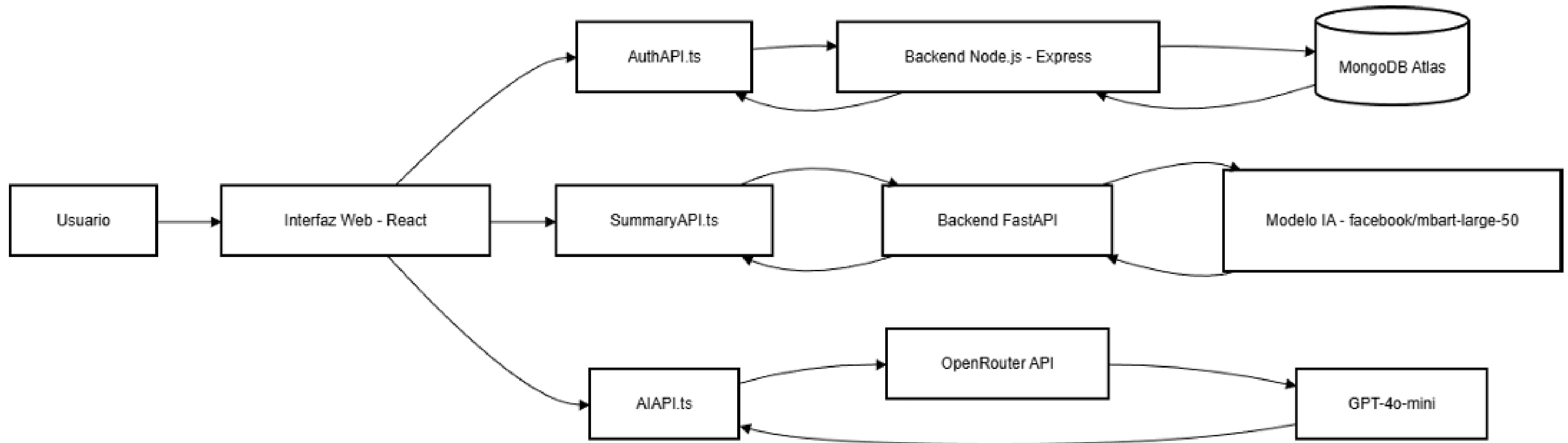


DIAGRAMA DE BLOQUES

COMPONENTES PRINCIPALES



Frontend

Es la interfaz de usuario del sistema. Está desarrollada en React y permite al usuario

- Registrarse e iniciar sesión
- Ingresar textos a resumir

Además, el frontend cuenta con una API interna encargada de comunicarse con OpenRouter, que es el servicio utilizado para acceder a modelos avanzados como GPT-4.

- Visualizar los resultados generados por la IA
- Comunicarse tanto con el backend de autenticación como con el backend de IA.

COMPONENTES PRINCIPALES



Backend Autenticación

Este backend es responsable de gestionar todo lo relacionado con:

- Registro de usuarios.
- Inicio de sesión.
- Encriptación y verificación de contraseñas.

Se conecta directamente con la base de datos para almacenar y validar la información de los usuarios.

- Generación de tokens JWT para sesiones seguras.
- Envío de correos de confirmación y recuperación de contraseña mediante Nodemailer.

COMPONENTES PRINCIPALES



Base de Datos (MongoDB)

La base de datos almacena:

- Información de los usuarios.
- Correos electrónicos.
- Contraseñas encriptadas.

MongoDB fue elegido por su flexibilidad y buen rendimiento en aplicaciones modernas basadas en JSON.

- Tokens de recuperación y confirmación.
- Información necesaria para la gestión de sesiones.

COMPONENTES PRINCIPALES



Backend IA (FastAPI)

Backend independiente cuya función principal es la generación de resúmenes automáticos mediante Inteligencia Artificial.

Este backend se encarga de recibir textos enviados desde la interfaz web y procesarlos para producir un resumen coherente, reducido y estructurado.

se apoya en un modelo especializado en tareas de resumen automático, el cual es ejecutado localmente a través de la librería Transformers.

Este backend funciona independiente al sistema de autenticación, lo cual mejora la organización, la escalabilidad y el mantenimiento del sistema en general

COMPONENTES PRINCIPALES



OpenRouter + GPT-4

Este servicio se encarga de:

- Recibe las solicitudes del frontend.
- Procesa el texto utilizando modelos avanzados.

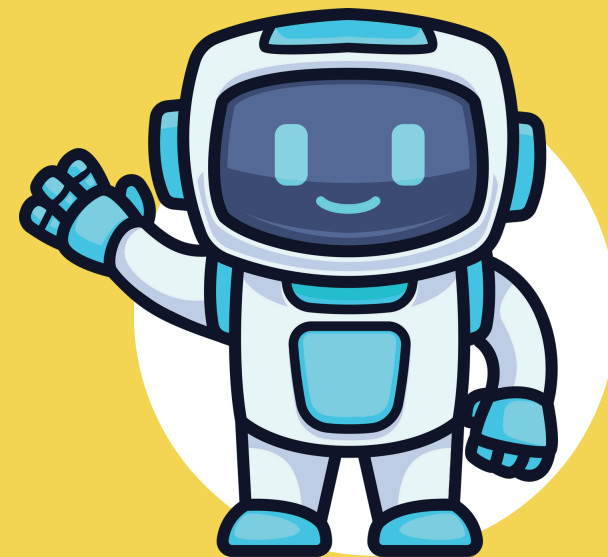
OpenRouter funciona como intermediario para acceder a modelos de lenguaje como GPT-4

- Devuelve el ensayo revisado, corregido y con sugerencias al sistema

RESUMEN DE ARQUITECTURA

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 de usuarios
IA Corrección de Ensayos	OpenRouter (GPT-4o-mini)	Revisión de ensayos, corrección y sugerencias
IA Creación de Resúmenes	FastAPI + Transformers (mbart-large-50)	Generación de resúmenes de texto
Despliegue	Render + Hugging Face Spaces	Despliegue

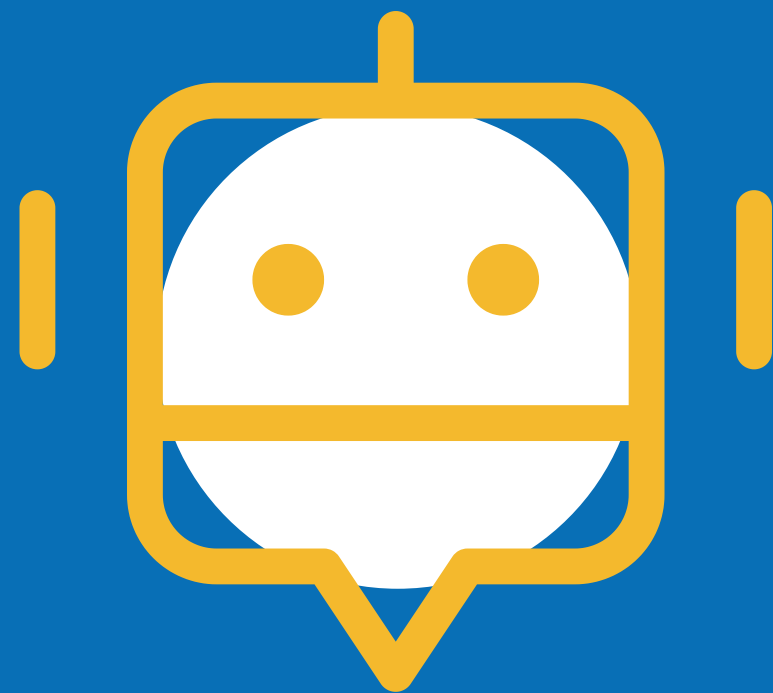
SISTEMAS INTELIGENTES



- Sistema capaz de analizar textos extensos en español
- Genera resúmenes coherentes y significativos
- Basada en un modelo tipo Transformer
- Utiliza el modelo `facebook/mbart-large-50`
- Modelo entrenado previamente para tareas de resumen en varios idiomas



SISTEMAS INTELIGENTES



Desarrollo e Implementación del Sistema:

- Integración de modelo preentrenado en una API propia con FastAPI
- Preprocesamiento y limpieza del texto de entrada
- Segmentación en fragmentos (chunks) para textos largos
- Control de longitud del resumen (mínimo y máximo)
- Segundo proceso de refinamiento del resumen final
- Validaciones para evitar errores y mejorar resultados
- Sistema optimizado para la generación automática de resúmenes

SISTEMAS INTELIGENTES



El backend en FastAPI:

- Recibe el texto enviado desde el frontend
- Procesa la información utilizando el modelo de IA basado en Transformers
- Devuelve un resumen optimizado al usuario

Tipo de resumen:

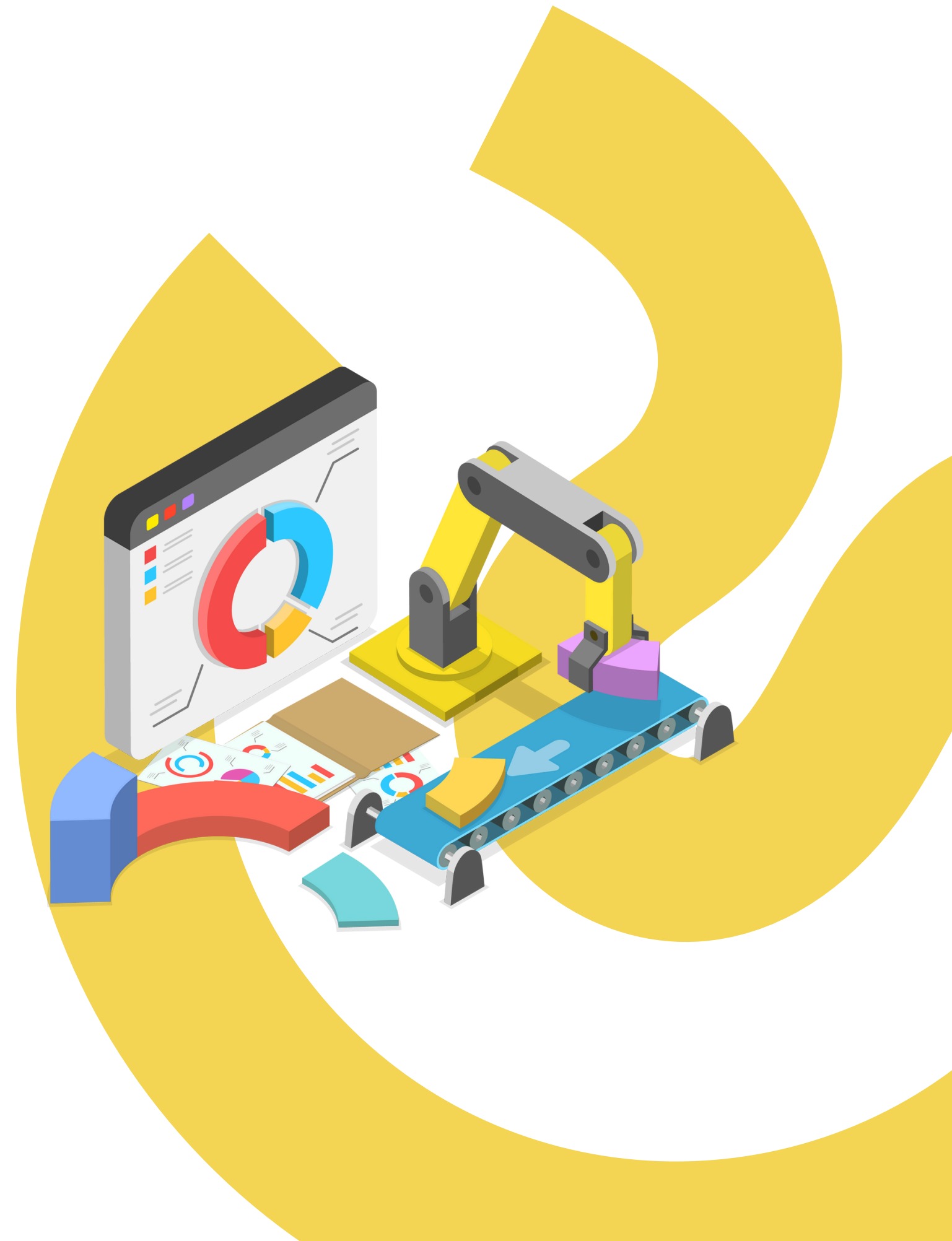
- Es un resumen abstractive
- No copia frases directamente del texto original
- Analiza el significado y genera nuevas oraciones
- Representa la idea principal del contenido

SISTEMAS DISTRIBUIDOS

BetterEssay utiliza una arquitectura distribuida, ya que sus componentes principales:

- Están separados física y lógicamente
- Se despliegan en distintas plataformas en la nube
- Se comunican mediante protocolos de red estandarizados

Esto permite dividir las responsabilidades del sistema en múltiples servicios independientes.



SISTEMAS DISTRIBUIDOS

El sistema está compuesto por los siguientes nodos:

- Frontend en React (Render)
- Backend en Node.js / Express (Render)
- Base de datos en MongoDB Atlas
- Microservicio IA en FastAPI (Hugging Face Spaces)
- Servicio externo de corrección (OpenRouter – GPT-4o Mini)

