Gaston Alasia
Matias della Torre
Micaela Perillo

Transformers

Trabajo Práctico Nro 2

Responsible AI y Safety

Se buscó aplicar los principios de *Responsible AI* para garantizar que el sistema desarrollado sea confiable, transparente y seguro.

Fairness

Los videos elegidos están curados. Solo fueron seleccionados aquellos que provienen de una fuente confiable como son los canales de Youtube de Harvard, Stanford entre otros, lo que evita que la información sea de baja calidad, maliciosa o falsa. Debido a que es un dataset variado y con una gran cantidad de videos (aproximadamente 7000), no se cuenta con biases hacia ningún tema o ninguna fuente en particular, por lo que el modelo considerará todas las fuentes de información y será justo.

El sistema informa al usuario por qué se le recomienda un determinado video, lo cual nos explica la decisión del modelo. De esta manera, el usuario puede comprender qué factores o elementos fueron considerados por el modelo durante la recomendación.

Robustness

Se espera que el modelo sea resistente a *prompt injections* y que el *pipeline* maneje adecuadamente los casos en los que no existan videos relevantes para retornar, evitando así la generación de respuestas incorrectas o alucinadas.

Transparency

Se explica abiertamente el funcionamiento del modelo, el origen del dataset y el preprocesamiento realizado a los datos. Además, el proyecto es open source, permitiendo la revisión del código por parte de terceros.

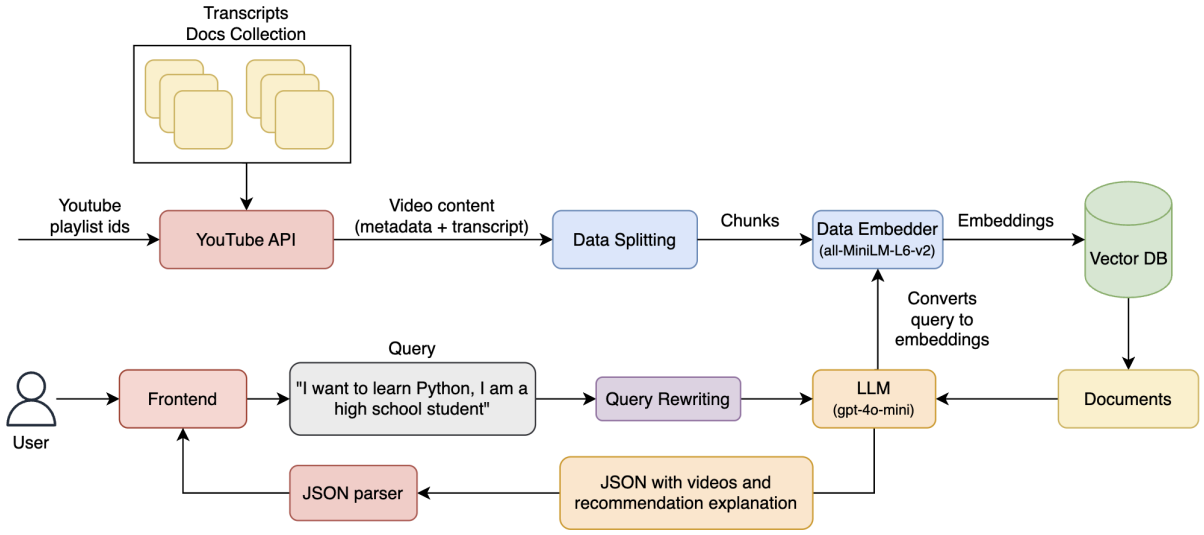
Data Privacy

No se almacenan datos personales de los usuarios. La información pedida al usuario es utilizada únicamente para generar el contexto que se le envía al modelo, pero no se persisten en ningún momento.

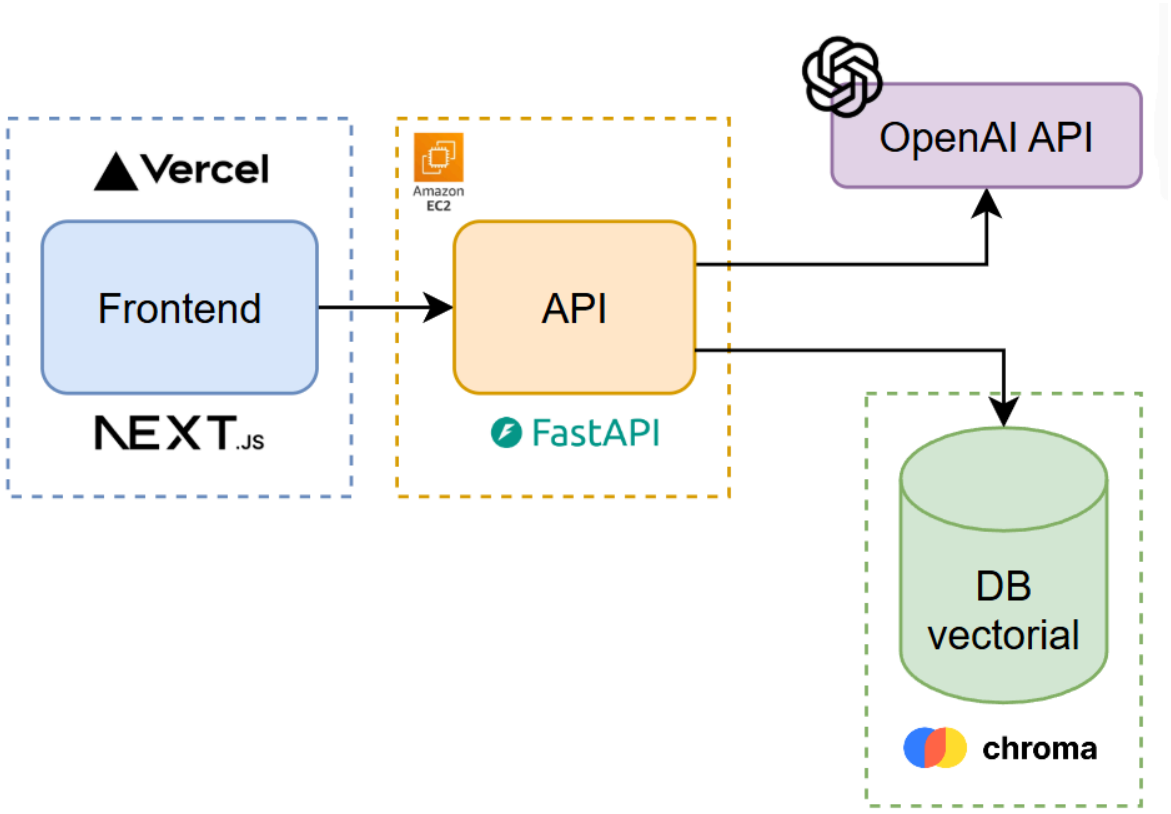
Gaston Alasia
Matias della Torre
Micaela Perillo

Producción

El pipeline de la aplicación es el siguiente



El usuario, al utilizarla, lo hace mediante esta arquitectura



El frontend fue desarrollado en Next.js, por lo que el deploy se hará en Vercel para aprovechar las facilidades que le brinda a dicho framework. La API, desarrollada en FastAPI, se ejecutará dentro de un contenedor Docker deployado en una instancia EC2 (Ubuntu). Por el lado de la Base de Datos vectorial, se encuentra hosteada en la propia cloud de Chroma (Chroma Cloud).