Stack

**Stack confirmado (actualizado)**

* **Lenguaje: Python.**
* **Frameworks:** 
  + **FastAPI (para endpoints).**
  + **Pandas (para manipulación y limpieza de datos).**
  + **Psycopg2 (para interactuar con Supabase PostgreSQL y PGVector).**
* **LLM/Embeddings:** 
  + **Hugging Face Transformers (modelo sentence-transformers como all-MiniLM-L6-v2 para embeddings).**
  + **Análisis de sentimiento:** 
    - **Hugging Face Transformers (modelo preentrenado distilbert-base-uncased-finetuned-sst-2-english).**
    - **VADER (análisis basado en reglas, rápido y efectivo para textos cortos).**
    - **TextBlob (análisis basado en patrones, simple y con polaridad/subjetividad).**
  + **Análisis de tópicos:** 
    - **Gensim (modelo LDA para extracción de tópicos).**
* **Almacenamiento vectorial: Supabase PGVector.**
* **Base de datos: Supabase PostgreSQL.**
* **Control de versiones: GitHub.**
* **Otras herramientas:** 
  + **Supabase Python Client (para interactuar con Supabase).**
  + **Matplotlib/Seaborn (para visualizaciones, como ranking de tópicos).**
  + **VS Code (entorno de desarrollo con scripts .py).**
  + **Scikit-learn (para preprocesamiento o utilidades, si es necesario).**
  + **Python-dotenv (para manejar credenciales).**

**Propuesta de funcionalidades**

Dado el límite de 48 horas y la instrucción de elegir 1 o 2 funcionalidades, propongo enfocarnos en:

1. **Generación y Guardado de Embeddings** (Funcionalidad 1):
   * **Por qué**: Es crítica para consultas RAG y análisis semántico, formando la base para futuros análisis. Implementar esto demuestra competencia en manejo de embeddings y almacenamiento vectorial, que son tecnologías avanzadas.
   * **Alcance**:
     + Procesar textos (mensajes, feedback, transcripciones).
     + Generar embeddings con sentence-transformers.
     + Guardar en Supabase PGVector con metadatos (userId, teamId, simulaciónId, tipo de interacción, timestamp).
     + Crear un endpoint FastAPI para verificar duplicados.
     + **Bonus**: Implementar limpieza básica de texto (por ejemplo, eliminar stopwords, normalizar texto).
   * **Impacto**: Muestra habilidades en integración con Supabase, manejo de embeddings y desarrollo de APIs.
2. **Análisis de Sentimiento y Tópicos Recurrentes** (Funcionalidad 2):
   * **Por qué**: Permite demostrar versatilidad con un modelo de ML preentrenado (Hugging Face) para análisis de sentimiento, combinado con una técnica simple como LDA para tópicos. Esto resalta tu capacidad para integrar enfoques modernos y escalables.
   * **Alcance**:
     + Procesar texto libre (feedback, transcripciones).
     + Analizar sentimiento con un modelo preentrenado de Hugging Face (distilbert-base-uncased-finetuned-sst-2-english).
     + Extraer tópicos con LDA (Gensim).
     + Guardar resultados en Supabase PostgreSQL (userId, simulaciónId, tipo de texto, resultado).
     + **Bonus**: Visualizar un ranking de tópicos por equipo con Matplotlib/Seaborn.
   * **Impacto**: Destaca tu conocimiento en NLP, ML preentrenado, y visualización, alineándose con el objetivo de evaluar habilidades blandas.

**Razonamiento**:

* Elegir ambas funcionalidades cubre un espectro amplio: **Funcionalidad 1** muestra competencia en almacenamiento vectorial y APIs, mientras que **Funcionalidad 2** destaca análisis avanzado con ML y visualización.
* El modelo preentrenado de Hugging Face es rápido de implementar (no requiere entrenamiento) y añade un componente moderno que impresionará en la prueba técnica.
* Priorizamos los **bonuses** (limpieza de texto y ranking de tópicos) porque son alcanzables en 48 horas y añaden valor al entregable.

**Cómo manejaremos las comparaciones en la Funcionalidad 2**

Para la **Funcionalidad 2 (Análisis de Sentimiento y Tópicos Recurrentes)**, implementaremos el análisis de sentimiento con tres métodos (**Hugging Face**, **VADER**, **TextBlob**) y compararemos sus resultados. Esto se reflejará en el código, la documentación (README), y posiblemente en una visualización. Aquí está el plan:

1. **Análisis de sentimiento**:
   * **Hugging Face Transformer**:
     + Usaremos distilbert-base-uncased-finetuned-sst-2-english para clasificar textos en positivo/negativo (con probabilidad).
     + Ventaja: Alta precisión, especialmente en contextos complejos.
   * **VADER**:
     + Proporciona una puntuación compuesta (compound) que clasifica el texto como positivo, negativo o neutral.
     + Ventaja: Rápido, optimizado para textos cortos (como mensajes o feedback).
   * **TextBlob**:
     + Devuelve polaridad (de -1 a 1) y subjetividad (de 0 a 1).
     + Ventaja: Simple, incluye subjetividad para evaluar cuán opinativo es el texto.
   * **Comparación**:
     + Procesaremos los mismos textos (por ejemplo, feedback o transcripciones) con los tres métodos.
     + Guardaremos los resultados en Supabase PostgreSQL con columnas para cada método (por ejemplo, hf\_sentiment, vader\_compound, textblob\_polarity).
     + Generaremos un resumen comparativo (por ejemplo, una tabla en el README o una visualización con Matplotlib/Seaborn) mostrando las diferencias en clasificación.
2. **Análisis de tópicos**:
   * Usaremos **LDA con Gensim** para extraer tópicos recurrentes (sin cambios aquí, ya que no mencionaste alternativas para tópicos).
   * **Bonus**: Visualizaremos un ranking de tópicos por equipo con Matplotlib/Seaborn.
3. **Impacto en la prueba técnica**:
   * Comparar tres métodos de análisis de sentimiento demuestra un enfoque analítico profundo y versatilidad, lo cual será bien valorado.
   * Documentaremos en el README las fortalezas y limitaciones de cada método, justificando su inclusión.

Supabase: kjfkmdklfmgslñmñlsmf´ls6464654sdkksdf

Supabase url :

NEXT\_PUBLIC\_SUPABASE\_URL=https://icxgavjswwlwnshvnhga.supabase.co

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImljeGdhdmpzd3dsd25zaHZuaGdhIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDY2MjgwNjIsImV4cCI6MjA2MjIwNDA2Mn0.jcqUd8YVfVdaQtg1w0apdLSjx3MReJNjnk4HkutONlM

**2. Explicación de la tabla sentiment\_topics**

La tabla sentiment\_topics almacenará los resultados del análisis de sentimiento y tópicos para cada texto procesado (mensajes, feedback, transcripciones), complementando la tabla embeddings (que almacena representaciones vectoriales para búsquedas semánticas). Cada columna tiene un propósito específico en el contexto del proyecto, especialmente para evaluar habilidades blandas (resiliencia, colaboración, etc.) y cumplir con los objetivos de la Funcionalidad 2. A continuación, explico cada columna y su significado:

* **id (SERIAL, PRIMARY KEY)**:
  + **Qué es**: Un identificador único para cada registro en la tabla.
  + **Significado**: Permite distinguir cada análisis de sentimiento/tópicos y facilita consultas o uniones con otras tablas (por ejemplo, relacionar con embeddings).
  + **Uso**: Indexación interna para rastrear registros.
* **userId (VARCHAR(50))**:
  + **Qué es**: El identificador del usuario que generó el texto (por ejemplo, "user123").
  + **Significado**: Vincula el análisis a un individuo, permitiendo evaluar sus actitudes o habilidades blandas (por ejemplo, resiliencia al analizar sentimientos consistentes en userId).
  + **Uso**: Filtrar interacciones por usuario para análisis personalizados (por ejemplo, "cómo se comporta user123 en simulaciones").
* **teamId (VARCHAR(50))**:
  + **Qué es**: El identificador del equipo al que pertenece el usuario (por ejemplo, "teamA").
  + **Significado**: Contextualiza las interacciones dentro de un equipo, útil para evaluar colaboración o compañerismo (por ejemplo, sentimientos positivos en interacciones de teamId).
  + **Uso**: Agrupar datos por equipo para análisis grupales.
* **simulationId (VARCHAR(50))**:
  + **Qué es**: El identificador de la simulación o contexto en el que ocurrió la interacción (por ejemplo, "sim2025").
  + **Significado**: Permite analizar el comportamiento en contextos específicos (por ejemplo, una simulación de alta presión puede revelar resiliencia).
  + **Uso**: Filtrar datos por simulación para comparar actitudes en diferentes escenarios.
* **type (VARCHAR(50))**:
  + **Qué es**: El tipo de interacción (por ejemplo, "chat", "feedback", "reunión").
  + **Significado**: Clasifica la fuente del texto, ayudando a interpretar el contexto (por ejemplo, "feedback" puede indicar proactividad, "reunión" liderazgo).
  + **Uso**: Analizar patrones según el tipo de interacción.
* **text (TEXT)**:
  + **Qué es**: El texto original o limpio analizado (por ejemplo, "Gran trabajo en el proyecto").
  + **Significado**: Almacena el contenido para referencia o reanálisis, permitiendo correlacionar los resultados de sentimiento con el texto original.
  + **Uso**: Verificar manualmente el contexto de los resultados o usar en análisis posteriores.
* **emoticons (TEXT)**:
  + **Qué es**: Los emoticones extraídos del texto (por ejemplo, "😊 🥳").
  + **Significado**: Captura expresiones emocionales que pueden reforzar el análisis de sentimiento (por ejemplo, 😊 sugiere positividad, complementando hf\_sentiment).
  + **Uso**: Evaluar actitudes (como compañerismo) basadas en emoticones, especialmente en chats informales.
* **hf\_sentiment (VARCHAR(20))**:
  + **Qué es**: El sentimiento predicho por un modelo de Hugging Face (por ejemplo, "positive", "negative", "neutral").
  + **Significado**: Indica el tono emocional del texto, clave para evaluar habilidades blandas (por ejemplo, un hf\_sentiment="positive" en feedback sugiere resiliencia o colaboración).
  + **Uso**: Analizar tendencias emocionales por userId o teamId.
* **hf\_score (FLOAT)**:
  + **Qué es**: La probabilidad o confianza del sentimiento predicho por Hugging Face (por ejemplo, 0.85 para "positive").
  + **Significado**: Mide la certeza del modelo, ayudando a priorizar resultados confiables (por ejemplo, un hf\_score alto refuerza la validez del sentimiento).
  + **Uso**: Filtrar resultados con alta confianza para análisis más precisos.
* **vader\_compound (FLOAT)**:
  + **Qué es**: El puntaje compuesto de VADER (de -1 a 1), donde valores positivos indican sentimientos positivos y negativos indican lo contrario.
  + **Significado**: Proporciona una medida alternativa de sentimiento, sensible a emoticones y jerga (por ejemplo, un vader\_compound de 0.7 sugiere una actitud positiva).
  + **Uso**: Comparar con hf\_sentiment para validar resultados o detectar matices emocionales.
* **textblob\_polarity (FLOAT)**:
  + **Qué es**: La polaridad de TextBlob (de -1 a 1), donde valores positivos indican sentimientos positivos.
  + **Significado**: Ofrece otra perspectiva de sentimiento, útil para textos en español (por ejemplo, un textblob\_polarity de 0.5 indica optimismo).
  + **Uso**: Combinar con otros puntajes para un análisis más robusto.
* **textblob\_subjectivity (FLOAT)**:
  + **Qué es**: La subjetividad de TextBlob (de 0 a 1), donde valores altos indican opiniones subjetivas y bajos indican objetividad.
  + **Significado**: Revela si el texto es factual o emocional, útil para evaluar comunicación asertiva (por ejemplo, baja subjetividad en reuniones sugiere claridad).
  + **Uso**: Identificar patrones de objetividad en interacciones.
* **topics (TEXT)**:
  + **Qué es**: Los tópicos extraídos del texto usando LDA (por ejemplo, "cooperación, estrategia").
  + **Significado**: Identifica temas clave en las interacciones, ayudando a evaluar habilidades como liderazgo (tópicos como "estrategia") o manejo de conflictos ("resolución").
  + **Uso**: Analizar patrones temáticos por userId o simulationId.
* **timestamp (TIMESTAMP)**:
  + **Qué es**: La fecha y hora de la interacción.
  + **Significado**: Permite rastrear la evolución de actitudes a lo largo del tiempo (por ejemplo, un aumento en vader\_compound tras una capacitación indica mejora en resiliencia).
  + **Uso**: Analizar tendencias temporales o filtrar datos por período.

**Relación con el proyecto**:

* La tabla sentiment\_topics complementa la tabla embeddings (Funcionalidad 1) al proporcionar métricas de sentimiento y tópicos que permiten evaluar habilidades blandas (resiliencia, colaboración, liderazgo, etc.), como discutimos previamente.
* Por ejemplo, un usuario con muchos registros en sentiment\_topics con hf\_sentiment="positive", vader\_compound alto, y tópicos como "apoyo" en type="feedback" mostrará compañerismo y proactividad.
* Los emoticones almacenados en emoticons refuerzan el análisis emocional (por ejemplo, 😊 en un texto con hf\_sentiment="positive" confirma una actitud positiva).
* Esta tabla será poblada en la Funcionalidad 2, pero crearla ahora asegura que esté lista para cuando procesemos los textos con modelos de sentimiento (Hugging Face, VADER, TextBlob) y LDA.