Este es el informe general de desarrollo, donde iremos anotando nuestras experiencias al momento de desarrollar **LA EVALUACION**

Primero identificamos una empresa, en este caso usaremos **Everlast Chile** ya que conozco sobre cosas de deporte en general.

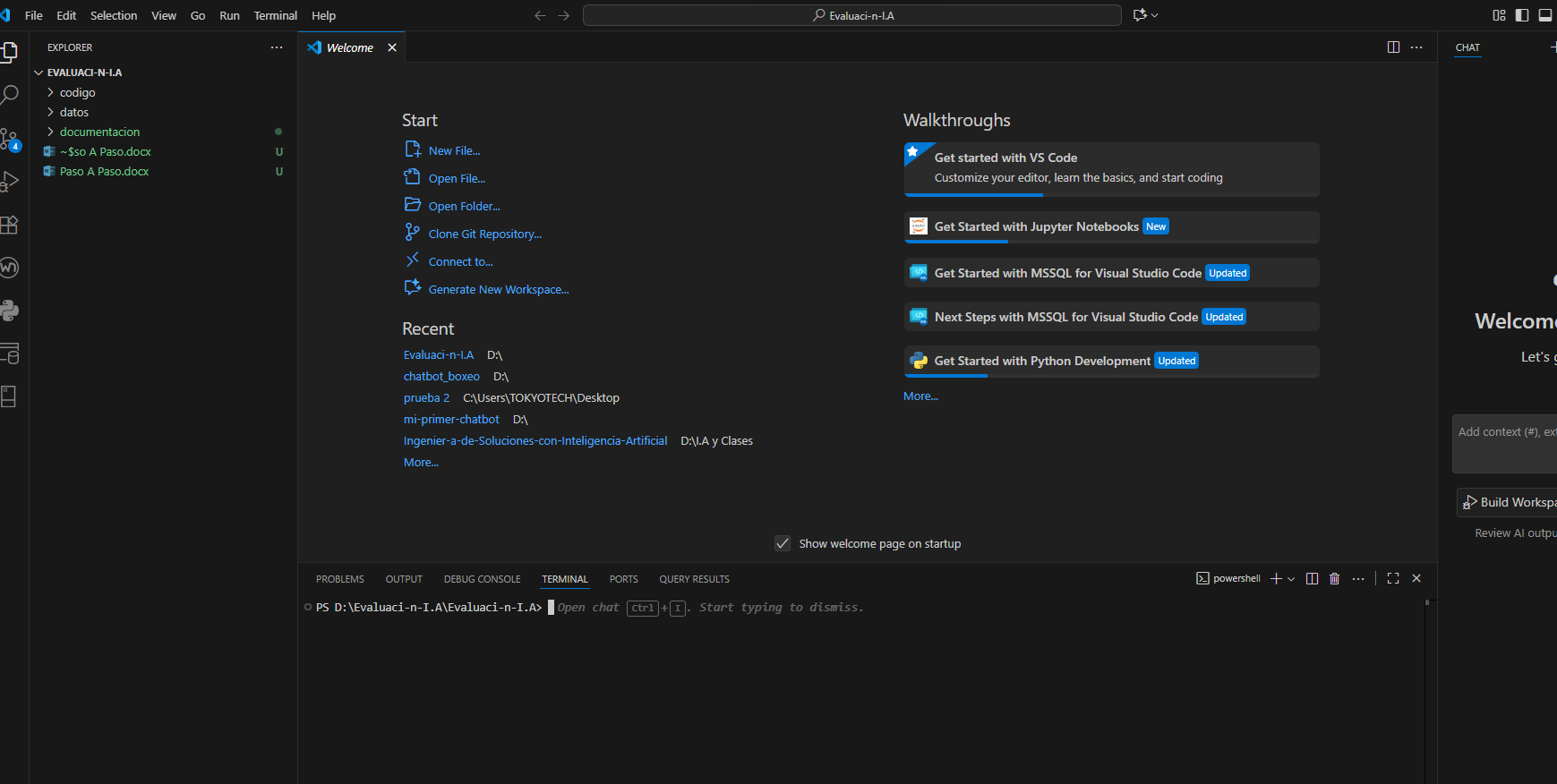
**Justificación:** Es una marca real, conocida y simple. Su catálogo de productos (guantes de boxeo, sacos, ropa) es específico y la información está públicamente disponible, y aunque no podamos acceder a ella, crearemos la información sobre los artículos para que la i.a la ocupe al momento de responder y no “alucine”, esto para un sistema RAG, ya que podemos simular que usamos su data interna.

Creamos 3 carpetas, en código todo el código de Python, en documentación, como el objetivo, la elección de la empresa y demás, en datos, pondremos archivos de texto con la información de la empresa, toda la información como de los artículos, las políticas y demás, para que nos sirva de simulación.

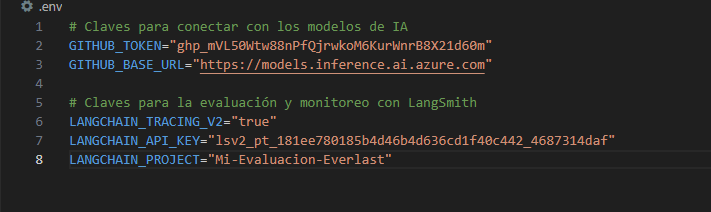
**AHORA COMENZAREMOS CON EL DESARROLLO DE CODIGO E IMPORTACIÓN DE HERRAMIENTAS**

**Partiremos creando el entorno virtual, y el archivo .env que nos sirve para mantener las llaves de acceso seguras.**

**Añadimos el archivo .env.**

****

****

****

**Adjuntamos nuestra key de github, y la api key de langsmith que nos servirá para el tema de las métricas.**

**Creamos el entorno virtual para que no se combinen las librerías y no haya problemas, asi podremos trabajar sin miedo y cómodamente.**

****

**Activamos el entorno.**

****

**Instalaremos las dependencias necesarias para poder trabajar, instalando las librerías necesarias para trabajar en este proyecto.**

****

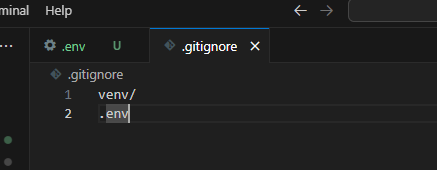
**También instalaremos la de lansgmith que a decir verdad, no sé si se instaló con la anterior entrada de comandos, creo que no :**

****

**También instalamo pip install langchain-community.**

**Hasta ahora con esta base, haremos nuestro primer commit.**

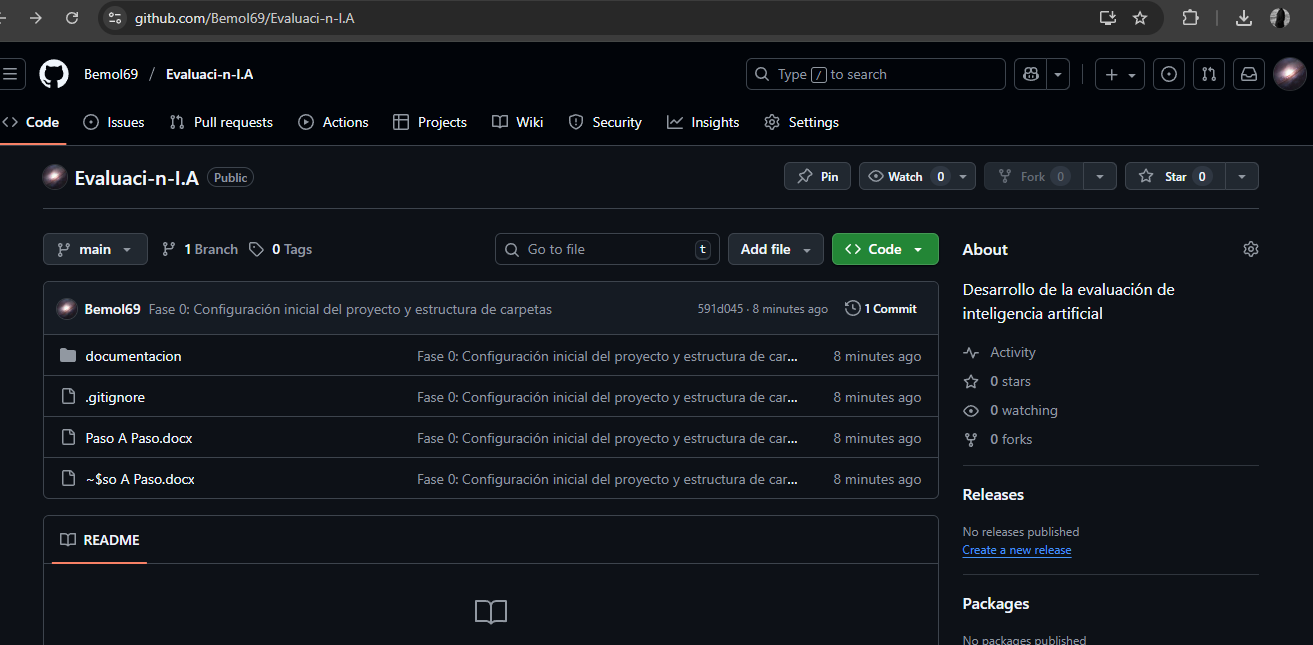
**Antes, como seguridad, crearemos un archivo .gitignore poniendo los archivos de venv y ,env de entorno para que no haga commit de eso, como regla de seguridad.**

****

**git add .**

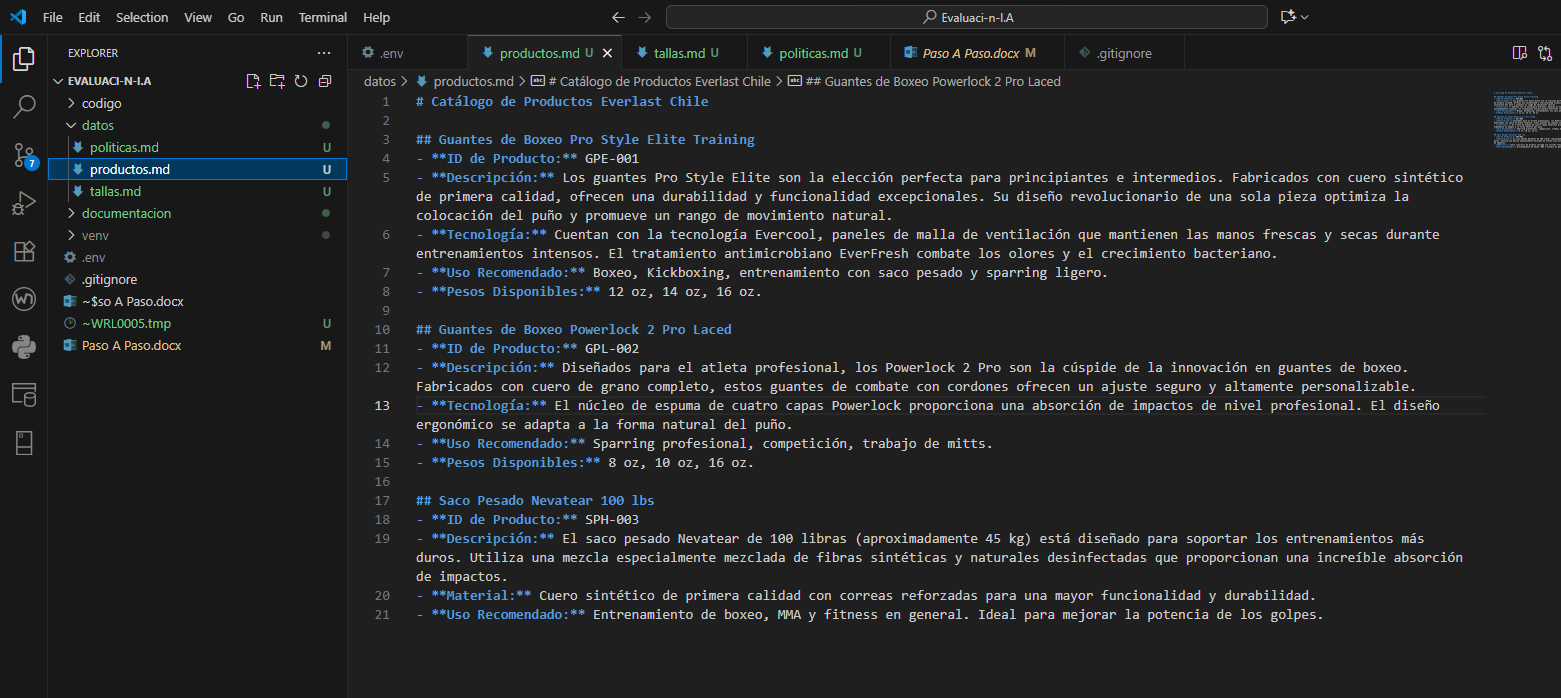
**git commit -m "Fase 0: Configuración inicial del proyecto y estructura de carpetas"**

**git push (a mi repositorio)**

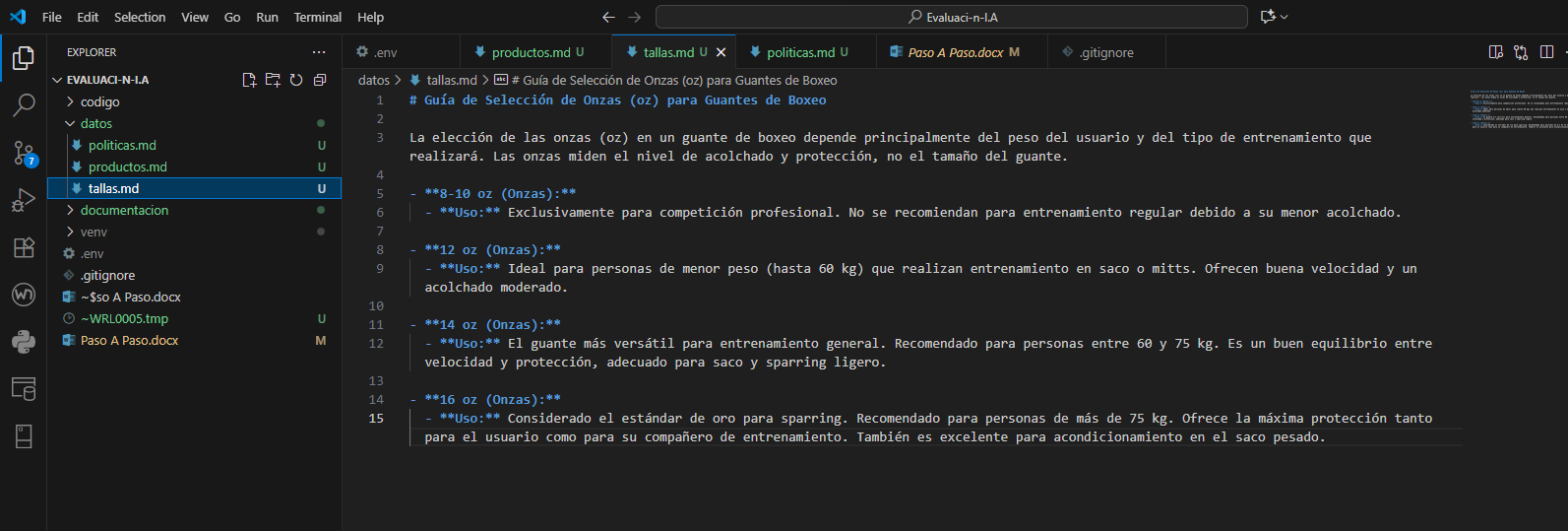
****

**Ahora lo siguiente que haremos es proporcionarle información a nuestra i.a, esto para que funcione el sistema RAG, crearemos la “base de conocimiento”, esto nos servirá para que nuestro asistente virtual de respuestas precisas y no alucine, partiremos creando 3 archivos dentro de la carpeta de datos, estos archivos serán:**

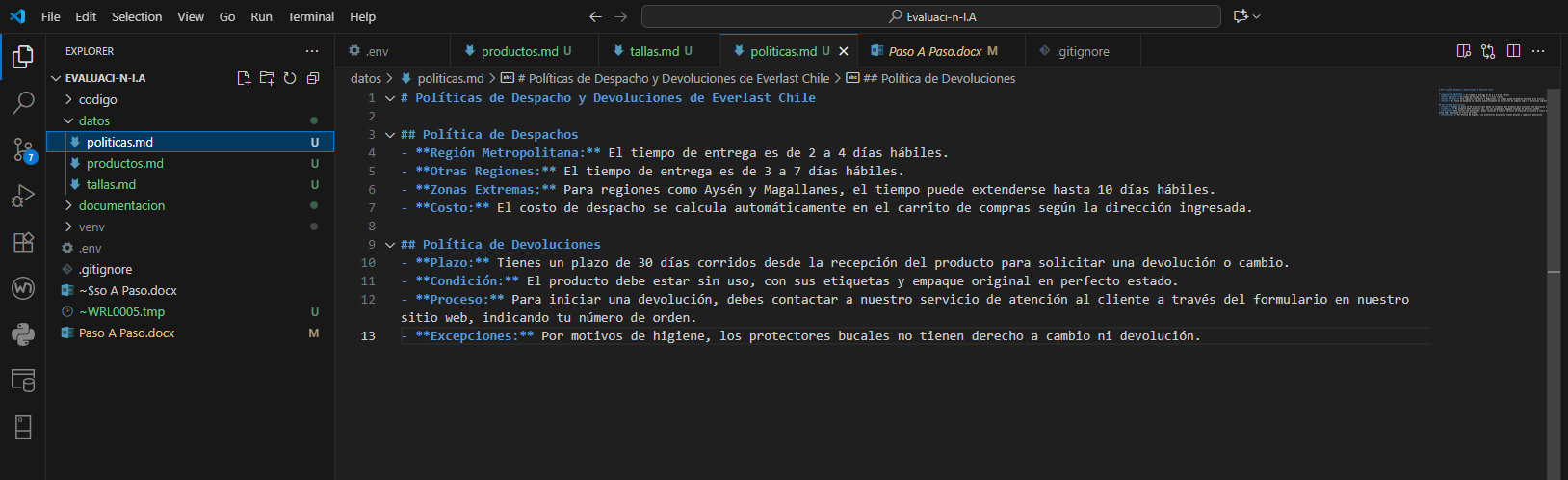
1. **productos.md (Este archivo contendrá la descripción de algunos productos clave de Everlast.)**

****

1. **tallas.md (Este documento ayudará al chatbot a responder una de las preguntas más frecuentes)**

****

1. **politicas.md (información de devoluciones y despachos.)**

****

**CREAREMOS OTRO COMMIT PARA SUBIR LA BASE DE CONOCIMIENTO A GITHUB**

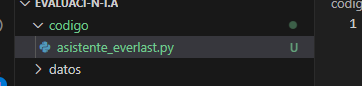
**Git add. Git commit, etc.**

**Ahora agregaremos la i.a con la que vamos a trabajar, en este caso ocuparemos el modelo gpt4o**

**Ocuparemos esta key de gpt4 : ghp\_acjSYV9PKW8GSwWkE7N02sq3zxxxxxxxxxxxxxxxxxxxxx**

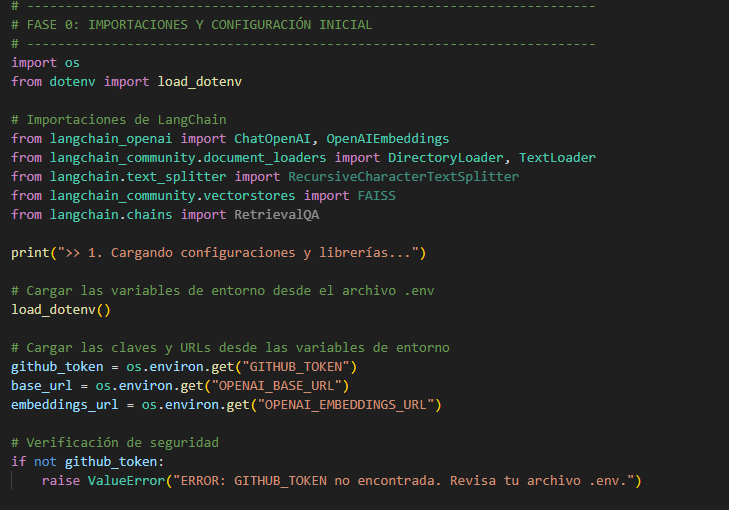
**Comenzaremos a generar el código en Python para desarrollar el chatbot:**

**Dentro de la carpeta código, crearemos el asistente asistente\_everlast.py**

****

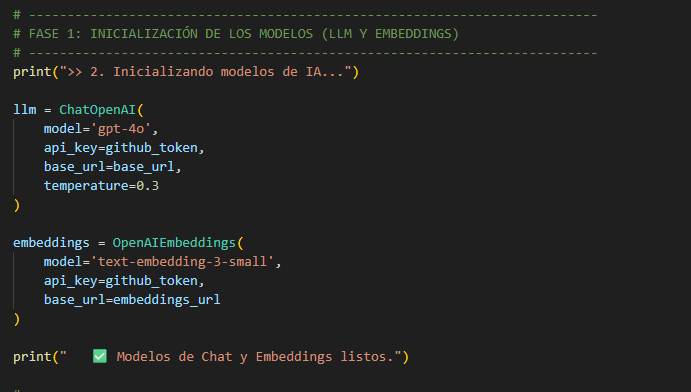
**Luego ocuparemos un código y este fue dividido en fases para ser mas fácil de reconocer sus partes y de que sirven.**

**Fase 0: Importamos todas las herramientas que necesitaremos de os, dotenv y langchain.**

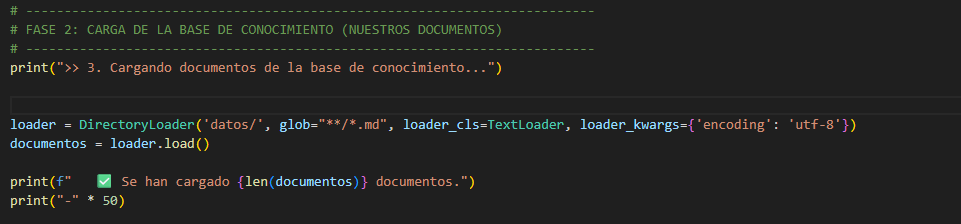
****

**Fase 1: Inicializamos los dos modelos de IA clave:**

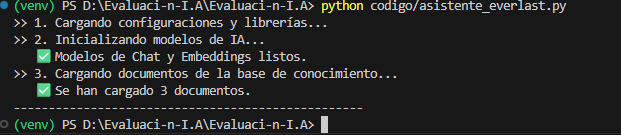
* **llm: El "cerebro" que conversará, usando la URL principal.**
* **embeddings: La "traductora" que convierte texto a vectores, usando la nueva URL de embeddings que definiste.**

****

**Fase 2: Usamos DirectoryLoader de LangChain, una herramienta súper útil que va a nuestra carpeta /datos y carga automáticamente todos los archivos .md que creamos en el paso anterior.**

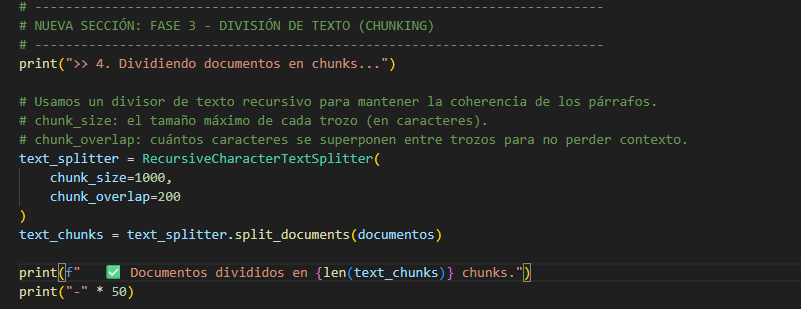
****

**ÉXITO**

****

**Esto nos muestra que nuestra base de datos esta establecida y funcinal… Seguimos.**

**Ahora crearemos los “CHUNKS” para la optimización y legibilidad de la base de conocimiento, para que la ia pueda trabar sin tantas dificultades.**

****

****

**Commit hasta la creación de chunks (se verán reflejados en los comits del repositorio)**