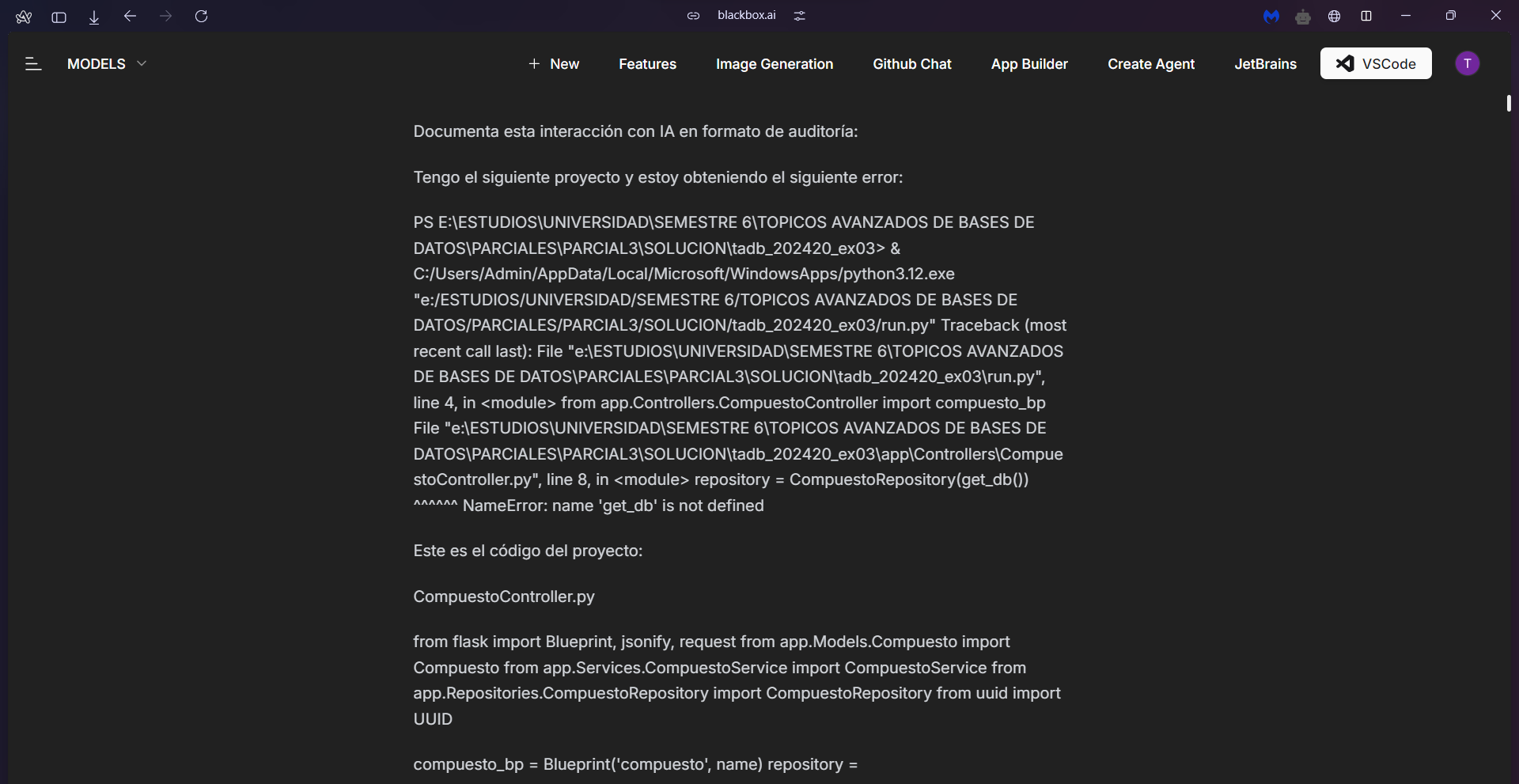
**DOCUMENTACIÓN IA**

Nota: tuve que pedirle a Blackbox.ai que documentara la interacción que realicé para solucionar este parcial debido a que era muy largo y eran demasiadas capturas.

****

* Finalmente, esto fue lo que documentó dicha IA (si lo documentó bien):

*“Contexto*

El usuario está trabajando en un proyecto de Flask que involucra interacciones con una base de datos PostgreSQL. Durante el desarrollo, el usuario experimenta un error relacionado con la función get\_db() que no está definida en el contexto de su aplicación. Este error es común en aplicaciones Flask cuando se intenta acceder a recursos que dependen del contexto de la aplicación sin haber establecido dicho contexto previamente.

*Descripción del Problema*

El usuario reporta el siguiente error al intentar ejecutar su aplicación Flask: "NameError: name 'get\_db' is not defined". Este error ocurre en el archivo CompuestoController.py, específicamente al intentar crear una instancia de CompuestoRepository utilizando get\_db(). La función get\_db() es crucial para obtener una conexión a la base de datos, y su ausencia indica que el código no puede acceder a los recursos necesarios para interactuar con la base de datos.

*Código Proporcionado*

El usuario proporciona el código de varios controladores y repositorios, así como la estructura de su aplicación. Se destaca la falta de importación de get\_db() en CompuestoController.py.

*Archivos Relevantes*

En el archivo CompuestoController.py, se observa que el usuario ha importado varias dependencias necesarias para el funcionamiento del controlador, pero no ha importado get\_db(). Esto es fundamental, ya que get\_db() es la función que permite establecer una conexión a la base de datos.

*Análisis del Error*

El error "NameError" indica que Python no puede encontrar la definición de get\_db() en el momento en que se intenta ejecutar la línea que crea el repositorio. Esto sugiere que la función no ha sido importada correctamente o que no está definida en el alcance adecuado.

*Posibles Causas del Error*

Falta de Importación: get\_db() no ha sido importada en CompuestoController.py.

Definición Incorrecta: get\_db() no está definida en el archivo database.py de la manera esperada.

Contexto de Aplicación: La función get\_db() debe ser llamada dentro de un contexto de aplicación válido para funcionar correctamente.

*Solución Propuesta*

Importar get\_db: Se sugiere importar get\_db en CompuestoController.py para que esté disponible en el contexto de ese archivo. Esto se puede lograr añadiendo la línea correspondiente al inicio del archivo, asegurándose de que la ruta de importación sea correcta.

Verificar Definición de get\_db: Asegurarse de que get\_db() esté correctamente definido en el archivo database.py dentro de la carpeta config. La definición debe incluir la lógica necesaria para establecer una conexión a la base de datos y almacenar dicha conexión en el objeto global g de Flask.

Revisar el Contexto de Uso: Se recomienda que el acceso a get\_db() se realice dentro de un contexto de aplicación válido. En Flask, esto significa que las funciones que dependen de g deben ser llamadas dentro de un contexto de solicitud, como en las funciones de vista o en los métodos de los controladores.

*Error Posterior*

Después de realizar los cambios sugeridos, el usuario encuentra un nuevo error: "RuntimeError: Working outside of application context". Este error indica que se está intentando acceder a la variable global g de Flask fuera del contexto de la aplicación. Esto ocurre cuando se intenta usar get\_db() antes de que Flask haya establecido el contexto de la aplicación.

*Análisis del Nuevo Error*

El error "RuntimeError" sugiere que el código está intentando acceder a recursos que requieren un contexto de aplicación, pero este contexto no está disponible en el momento de la ejecución. Esto puede suceder si se intenta crear instancias de clases que dependen de g antes de que la aplicación Flask esté completamente inicializada.

*Solución Propuesta para el Nuevo Error*

Uso de before\_request: Se sugiere mover la creación de CompuestoRepository a un método before\_request dentro del blueprint compuesto\_bp. Esto asegura que la instancia se crea en el contexto adecuado, justo antes de que se maneje una solicitud. Al hacerlo, se garantiza que get\_db() se llame en el contexto correcto, permitiendo que la conexión a la base de datos se establezca sin problemas.

Creación del Repositorio en la Ruta: Alternativamente, el usuario puede optar por crear la instancia de CompuestoRepository directamente dentro de las funciones de vista que manejan las solicitudes. Esto asegura que cada vez que se maneje una solicitud, se esté trabajando dentro del contexto de la aplicación y se pueda acceder a get\_db() sin problemas.

Manejo de Errores: Es recomendable implementar un manejo de errores adecuado para capturar cualquier excepción que pueda surgir al intentar acceder a la base de datos. Esto puede incluir el uso de bloques try-except para manejar errores de conexión y proporcionar retroalimentación útil al usuario.”

Igualmente, acá están los pantallazos de los prompts que realicé:

