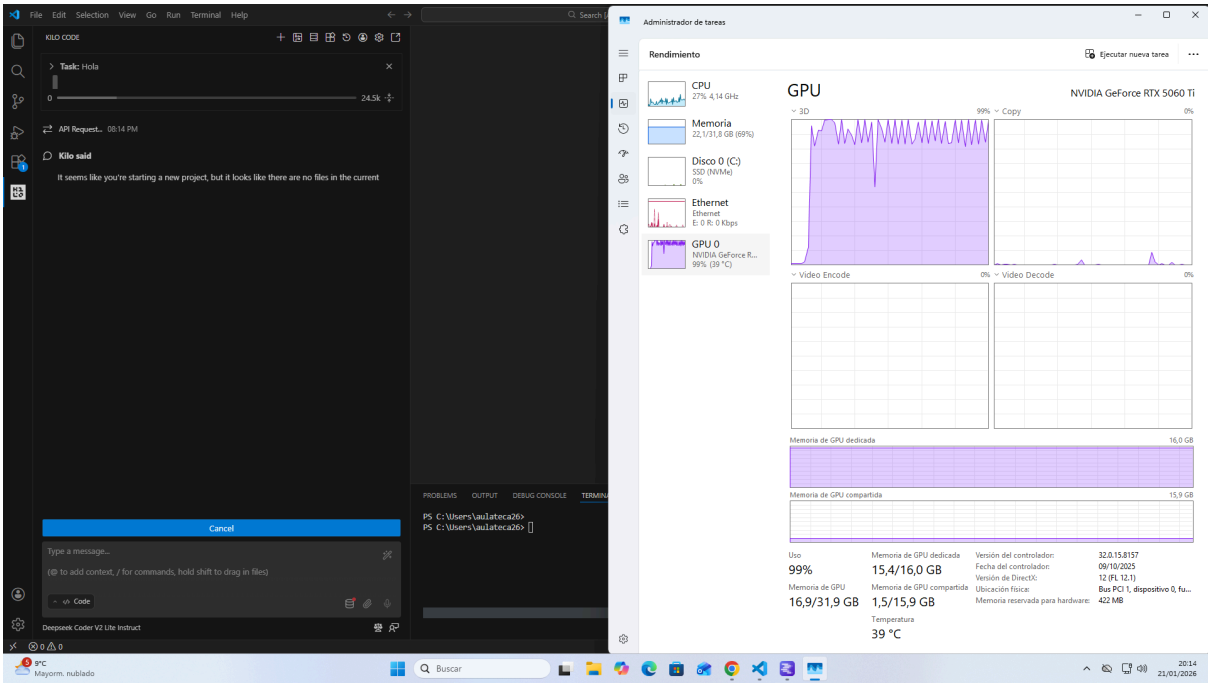
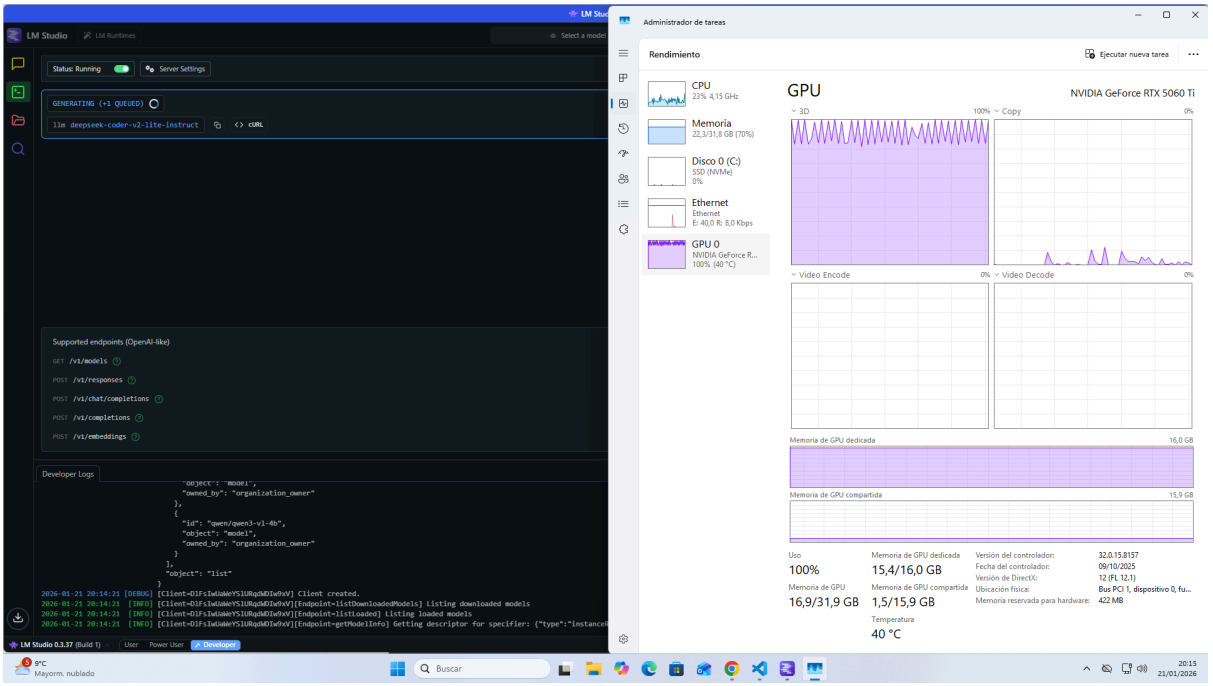
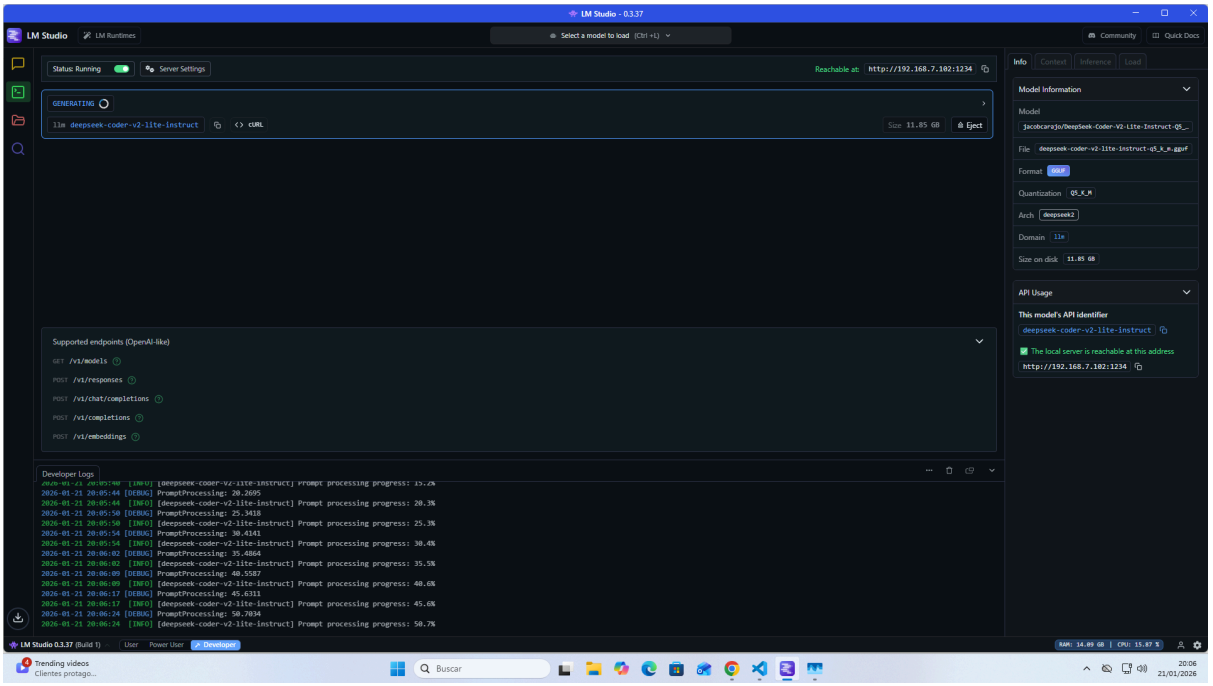


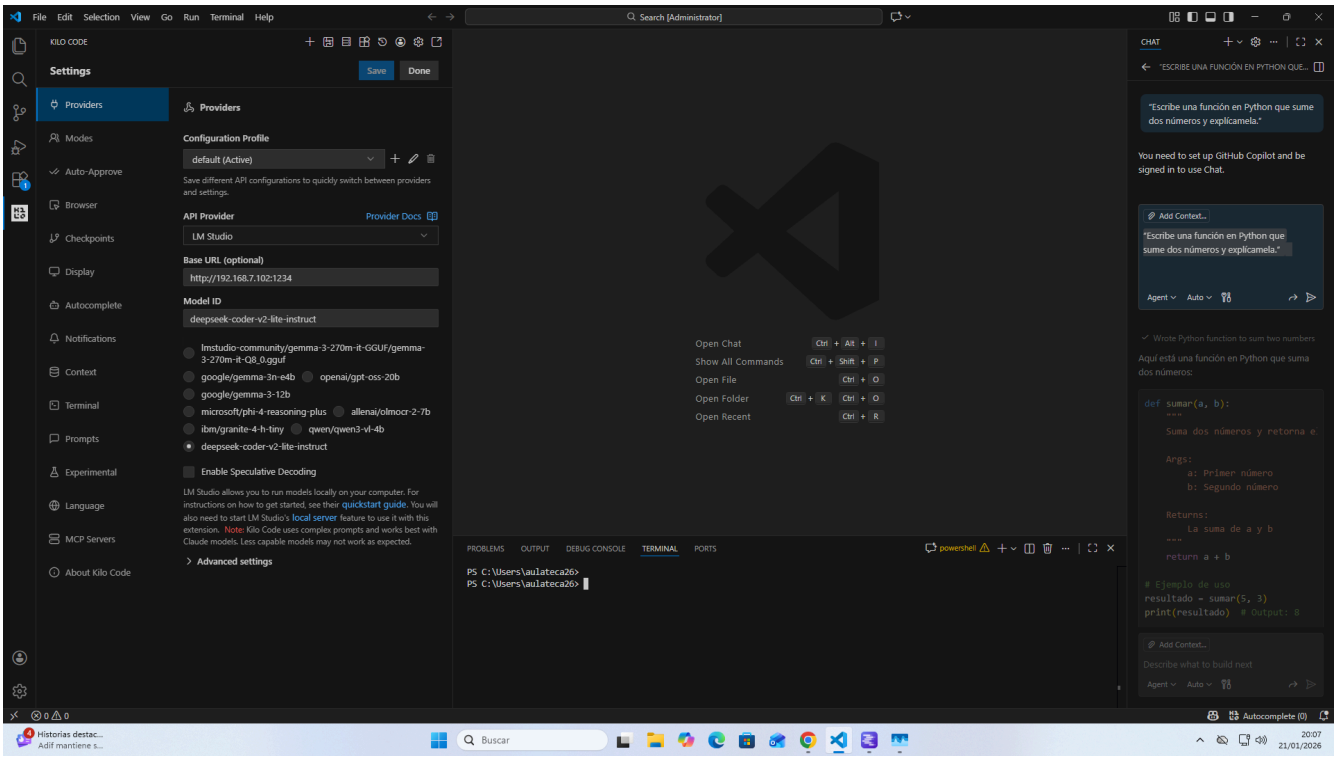
En esta imagen se observa el Administrador de tareas del sistema mientras el modelo de inteligencia artificial está en funcionamiento. La GPU aparece trabajando casi al máximo de su capacidad y la memoria de vídeo se encuentra prácticamente llena, lo que indica que el modelo está cargado completamente en la tarjeta gráfica. Esta situación demuestra que el procesamiento se realiza de forma local y que el equipo está ejecutando tareas de generación de texto en tiempo real. El alto uso de recursos refleja un proceso activo de inferencia, donde el modelo analiza el contenido recibido y produce una respuesta utilizando toda la potencia disponible del hardware.



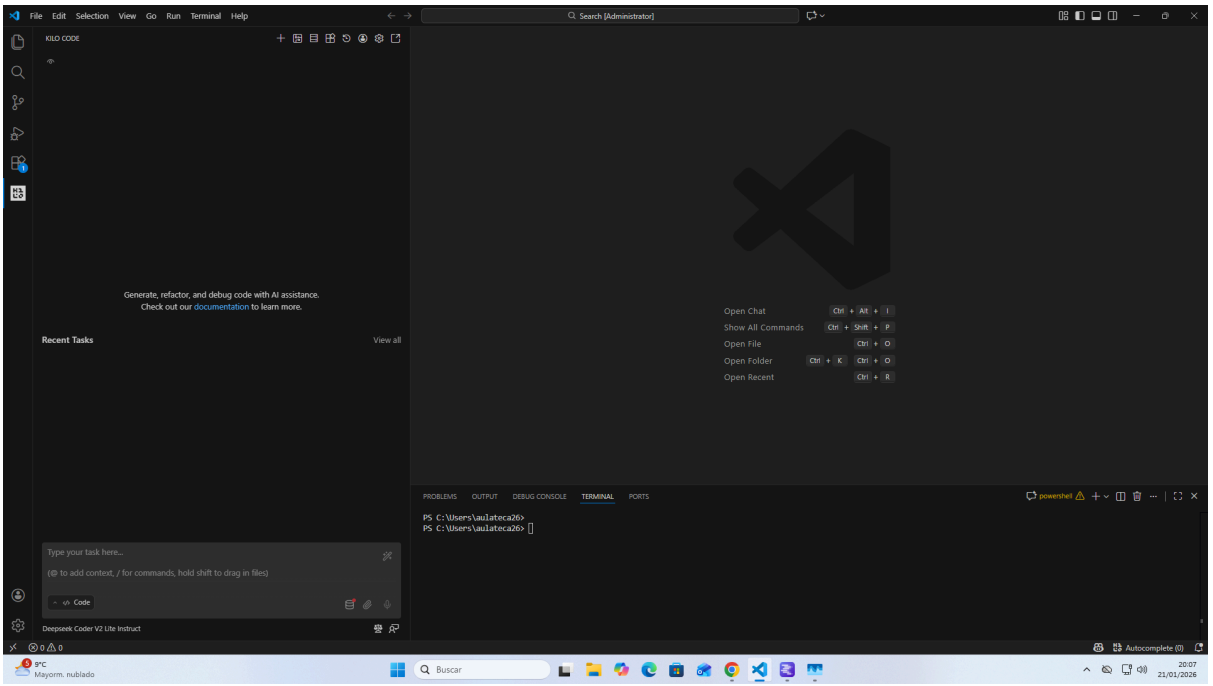
La segunda imagen muestra la interfaz del programa utilizado para ejecutar el modelo, donde se puede ver que el sistema se encuentra en estado activo o “Running”. Aparecen disponibles varios endpoints de API compatibles con el formato de OpenAI, lo que significa que otros programas pueden conectarse al modelo como si fuera un servicio externo. En esta situación el software actúa como un servidor local que permite enviar solicitudes y recibir respuestas desde diferentes aplicaciones dentro de la misma red. Esto convierte al modelo en un motor central que puede ser utilizado por herramientas de desarrollo o asistentes automáticos.



En esta captura se visualizan los registros internos del modelo durante el procesamiento del prompt. Se observan porcentajes de progreso que indican cómo el sistema analiza y prepara la información antes de comenzar a generar la respuesta final. Este paso corresponde a la conversión del texto en tokens y a la carga del contexto dentro de la memoria de la GPU. La situación refleja una fase previa a la generación, normalmente más lenta cuando el contenido enviado es largo o complejo, ya que el modelo necesita comprender todo el contexto antes de producir resultados coherentes.



La cuarta imagen presenta la configuración del agente dentro del entorno de desarrollo. Se muestran los parámetros necesarios para conectar el editor con el servidor local del modelo mediante una dirección IP y un puerto específico. Esto indica que el asistente de programación está enlazado directamente con el modelo ejecutado en el ordenador del usuario. En esta situación el editor actúa como interfaz de interacción, enviando instrucciones automáticamente al modelo y recibiendo sugerencias o código generado sin necesidad de conexión a servicios externos.



En la última captura se observa el uso práctico del agente dentro del editor, donde aparece una conversación solicitando la creación de una función en Python. El sistema responde utilizando el modelo local previamente configurado, mostrando cómo el flujo completo funciona de principio a fin. La situación representa la integración final del entorno: el usuario escribe una petición, el agente la envía al servidor local, el modelo procesa la información mediante la GPU y la respuesta vuelve al editor lista para ser utilizada. Esto confirma que el conjunto funciona como un asistente de programación autónomo ejecutado completamente en local.