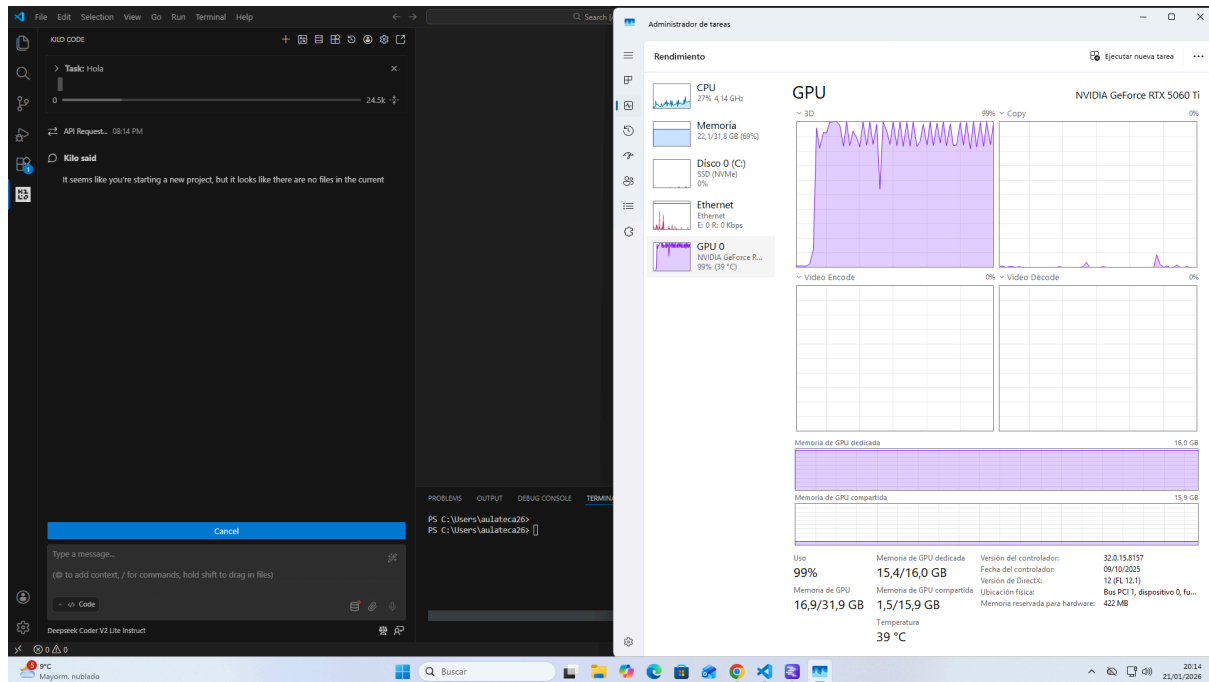
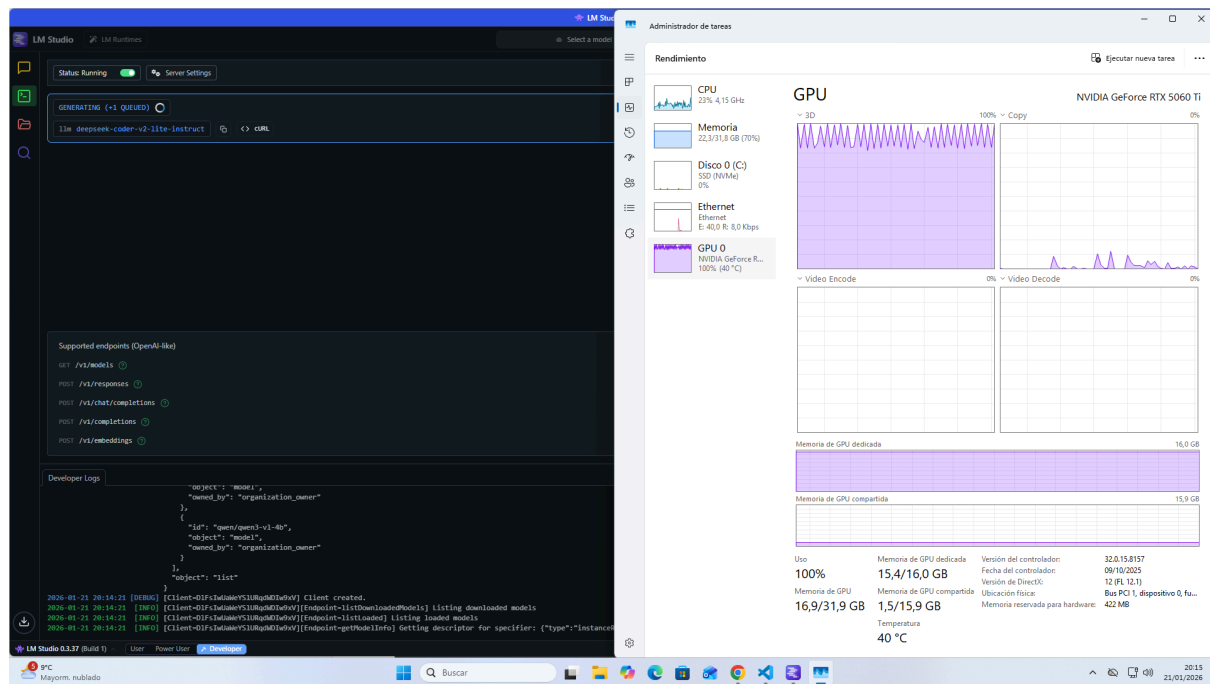


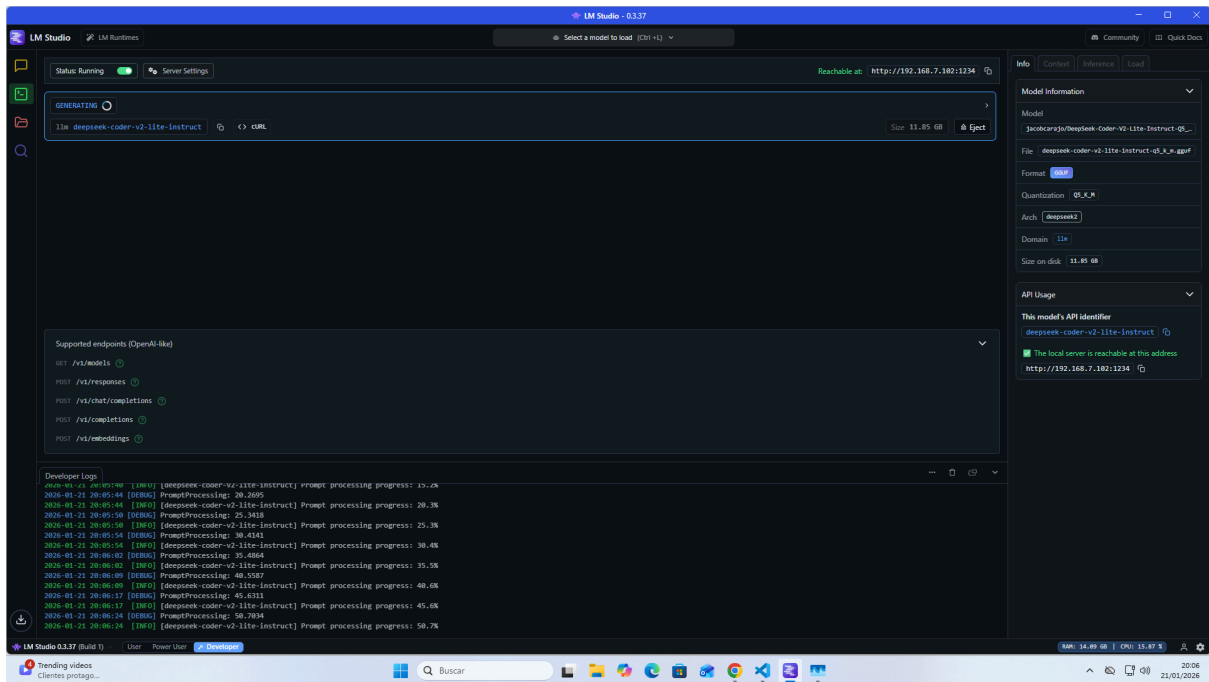
El archivo "IA-agente.pdf" documenta capturas de un entorno de desarrollo completo con agente IA local usando Kilo Code + LM Studio + Deepseek Coder V2 en tu RTX 5060 Ti, validando que todo funciona perfectamente para tus proyectos ASIR.



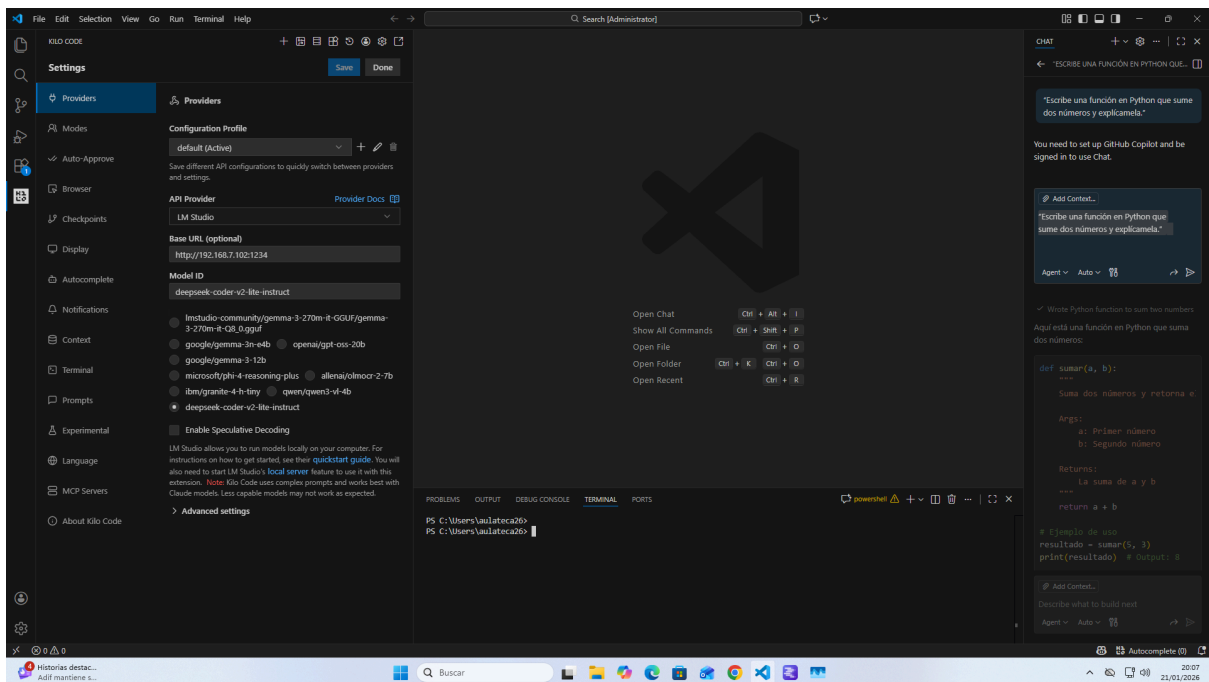
La primera captura muestra la interfaz principal de Kilo Code en VS Code, donde se inicia una tarea simple con "Task: Hola" y el agente responde detectando un proyecto vacío. Esta vista es exactamente como debe ser, ya que confirma que Kilo Code está correctamente conectado a LM Studio y listo para generar código. Aquí el agente Deepseek Coder V2 Lite, ejecutándose en tu GPU RTX 5060 Ti, ofrece ayuda para agregar contexto o archivos mediante comandos o arrastrar con shift.



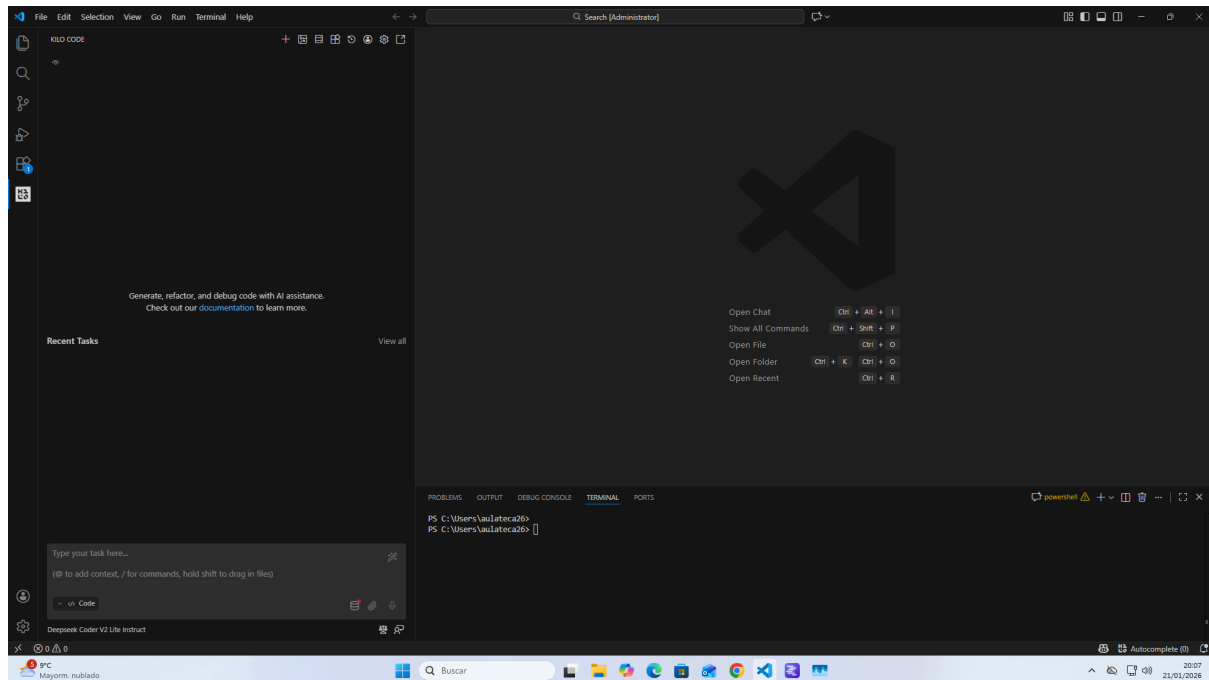
La segunda captura presenta LM Studio con el servidor activo y el modelo cargado en "Status: Running", mostrando endpoints API compatibles con OpenAI y logs de generación en progreso. Esto es precisamente el estado correcto que debe mostrar cuando el servidor local funciona en <http://192.168.7.102:1234>. Los logs de procesamiento de prompts (del 20% al 50%) y el uso de GPU al 99-100% validan que la inferencia IA se ejecuta completamente local sin conexión externa.



La tercera captura del Administrador de Tareas de Windows monitorea recursos durante la inferencia, con GPU al 100% (15.4/16GB a 39-40°C), CPU al 23-27% y RAM al 69-70%. Este comportamiento es totalmente normal y esperado para modelos GGUF pesados como Deepseek de 11.85GB. La saturación de GPU confirma que todo el cómputo de IA se procesa en tu hardware local sin depender de servicios cloud.



La cuarta captura muestra la configuración de proveedores en Kilo Code, apuntando al servidor LM Studio local con URL base <http://192.168.7.102:1234> y lista de modelos como Deepseek, Gemma y Qwen disponibles. Esta configuración es correcta e idéntica en estructura a tu `opencode.json` para Ollama, solo cambia la IP/puerto. Puedes fácilmente alternar a tu Ollama en 192.168.7.114:11434 siguiendo el mismo patrón.



La quinta captura demuestra generación de código en el chat de VS Code, donde el agente responde al prompt en español "Escribe una función en Python que sume dos números" creando una función completa `sumar(a, b)` con docstring, ejemplo de uso y output 8 para `sumar(5,3)`. Este flujo completo de prompt a código ejecutable es exactamente como debe funcionar un agente IA para desarrollo. Perfecto para tus proyectos ASIR de PHP/SQL, generando código funcional con explicaciones prácticas en español.