

L U M B R E



La IA también miente con confianza: alucinaciones en software

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

La IA te recomienda instalar un paquete que no existe, usa una función que nunca se escribió y cita una opción de API que nadie implementó —con la misma seguridad que lo demás—. No te miente: completa. Su naturaleza es generar lo plausible, y en software lo plausible incluye nombres que suenan reales. Este OA te explica por qué ocurre, las tres familias de alucinación que te muerden (dependencias, funciones/APIs, configuraciones), y el protocolo de verificación de 90 segundos que las atrapa.

QUÉ APRENDERÁS

- Reconocer por qué un modelo genera paquetes, funciones y APIs que no existen y cómo verificarlos.

No miente: completa



La IA no miente: completa. Este módulo entrena la desconfianza sana que atrapa dependencias inventadas, APIs fantasma y atajos peligrosos antes de que lleguen a producción.

Un modelo de lenguaje no consulta un catálogo de hechos: genera la continuación más plausible de lo que sabe. En prosa eso produce frases bonitas; en código produce **paquetes que suenan reales, funciones que nunca existieron y opciones de**

configuración lógicas pero falsas. Lo peligroso no es que a veces falle —falla en lo visible también—: es que la mentira plausible viene envuelta en la misma confianza que la verdad, sin ningún marcador que las distinga a simple vista.

ACTIVIDAD INTERACTIVA
DISPONIBLE EN LUMBRE

1. Las tres familias que muerden en software



Por qué la tercera es la más traicionera

La regla que nace de la tabla: lo que NO da error es lo que debes verificar con evidencia, no con confianza. Las dos primeras familias se autodenuncian; la tercera se disfraza de éxito.

Familia	Qué pasa al ejecutar	Ventana de detección
Dependencia inexistente	el instalador falla: «package not found». Ruidosa, inmediata	fácil: se atrapa sola... si no la instalas desde un script que la rebautiza
Función/API falsa	error «is not a function» o 404 de la API: ruidosa, al primer uso	media: se ve corriendo la función, no leyendo
Configuración inventada	NADA. El sistema ignora la clave desconocida y sigue como si nada	difícil: el «modo seguro» que creíste activar nunca se activó

El protocolo de verificación de 90 segundos

El protocolo convierte la desconfianza en tres movimientos mecánicos

Antes de aceptar cualquier código generado que mencione algo externo, verifica:

- PAQUETES (30 s):** busca el nombre exacto en el registro público (npm / PyPI / etc).
Copia el enlace de la página. Si no hay página, no hay paquete. Punto.
- FUNCIONES Y PARAMETROS DE API (40 s):** ve a la documentación OFICIAL del servicio (no al entrenamiento del modelo) y busca el nombre con ctrl-f. Si la documentación no lo menciona, trátalo como inexistente hasta probar lo contrario.
- CONFIGURACION (20 s):** ¿qué evidencia mostraría que esta opción SI está activa?
Pide esa evidencia: un log, un comportamiento observable, una consulta.

Regla de oro: la pregunta nunca es "suena correcto" — es "donde lo verifico".
Si no hay lugar donde verificar, no hay permiso para usarlo.

Para reforzar

Prueba

2. La IA configuró tu app con la opción «session_secure = true» en el archivo de ajustes. La ejecución no da ningún error. ¿Cuál es el procedimiento correcto?

- ☐ Está activa: si el sistema no protestó, la opción existe.
- ☐ Pedir evidencia observable de que el modo seguro está activo (un log, una consulta, un comportamiento) y verificarla.
- ☐ Cambiar el nombre a sessionSecure para que sea más estándar.
- ☐ Reiniciar el servidor para confirmar.

3. Si la IA escribió cien líneas correctas, la probabilidad de que la función que inventó en la línea 101 sea real es alta.

- ☐ Verdadero
- ☐ Falso

Detecta la instalación a ciegas

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN LUMBRE

**4. Un alumno cuenta su sesión con la IA.
Señala el momento en que ocurrió el
daño:**



5. Toma el último bloque de código que la IA te dio para tu proyecto y pásale el protocolo de 90 segundos completo. Anota cuántos nombres externos mencionaba y cuántos sobrevivieron a la verificación —el número de bajas es tu línea base de desconfianza productiva.

La confianza se construye verificando; la catástrofe se construye asumiendo.

Referencia curricular

Materia:	Desarrollo de Software con IA
-----------------	-------------------------------

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://empresa.lumbre.lat/object/fsi-cnf-0a01>

Material educativo generado desde Lumbre · lumbre.lat