

L U M B R E



El agente rompió algo: recuperarte con historia limpia

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

Llega la tarde en que el agente «mejora» algo que funcionaba y la app se cae. El pánico borra; el método recupera. Aquí practicas el protocolo de rescate: congelar el daño, localizar el último commit bueno con historial y mensajes, revertir hacia él con seguridad y escribir la lección en tu documento maestro para que no se repita. Perder código no es la catástrofe: perder la capacidad de volver es la catástrofe.

QUÉ APRENDERÁS

- Detectar la versión buena, revertir con seguridad y documentar la lección en el documento maestro.

El momento que estabas esperando (en negativo)

Pediste «optimiza las consultas» y el agente dejó la pantalla de pagos en blanco. Ya. ¿Ahora qué? Este OA es tu simulacro de incendio. La buena noticia: si has commiteado cada hito (M14-OA01), **siempre** hay una versión buena a la que volver. La recuperación no es talento: es protocolo.

ACTIVIDAD INTERACTIVA DISPONIBLE
EN LUMBRE

1. Protocolo de rescate en cuatro tiempos



Volver sin borrar la historia

Hay dos formas de regresar: la segura y la destructiva. **revert** crea una versión nueva que anula la mala: el historial cuenta todo, y si te equivocaste puedes volver a volver. Operaciones destructivas descartan commits de golpe y borran el rastro; úsalas solo sobre trabajo sin guardar o en tu propia rama, jamás sobre lo publicado.

El rescate se ve, no se esconde

```
$ git log --oneline -6                                # la línea de tiempo reciente\n
e4a190 (HEAD) "optimiza consultas" <-- sospechoso\n 7c2205 pagos funcionando\n
<-- la ultima version BUENA\n 1f8e33 etiqueta v1.2\n\n# pedir al agente la\n
reversa SEGURA (no borra historia):\n$ git revert e4a190                                #\n
crea un commit que anula el dano\n$ git log --oneline -2\n b9d4e0 Revert\n
"optimiza consultas"      # el dano queda documentado, no tapado\n$ git tag rescate-\n
v1.2.1                    # congelo el punto recuperado
```

La lección va al documento maestro

Recuperar el código es la mitad. La otra mitad: escribir en tu **STACK.md** (el documento maestro del curso, M05) qué orden causó la rotura, cómo se detectó y qué regla nueva rige de ahora en adelante. Ejemplo de regla: «toda optimización de consultas va en rama con pruebas antes de fusionarse». Sin la lección escrita, el mismo accidente se repite en seis semanas.

Prueba

2. El agente rompió pagos con un commit ya guardado. ¿Por qué se prefiere revert frente a una operación destructiva?

- ☐ Porque revert es más rápido.
- ☐ Porque revert crea una versión que anula el daño sin borrar el historial: se puede auditar y volver a volver.
- ☐ Porque revert borra el commit malo para que nadie lo vea.
- ☐ Porque revert no requiere haber commiteado antes.

3. Si hubieras commiteado cada hito que funcionaba, el incidente del agente se reduce a elegir la versión correcta y revertir.

- ☐ Verdadero
- ☐ Falso

Detecta el rescate mal ejecutado

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN
LUMBRE

4. Un alumno recupera su app tras un desastre. Señala la decisión que agravó el incidente:



5. Escribe hoy, en tu documento maestro, un protocolo de incendio para tu proyecto: dónde está tu última versión buena, cómo se revierte y qué regla nueva aprendiste esta semana. Léelo antes de tu próximo cambio grande.

Los profesionales no evitan todos los incendios: ensayan cómo salir de ellos.

Referencia curricular

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

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



<https://empresa.lumbre.lat/object/fsi-m14-0a03>

Material educativo generado desde Lumbre · lumbre.lat