

L U M B R E



El mapa del proyecto: qué archivo hace qué y dónde está la configuración

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

La IA no te entrega un archivo: te entrega una estructura de carpetas. Perderse ahí es la causa nº 1 de prompts inútiles («cambia lo de las facturas»... ¿dónde?). Todo proyecto se parece: pantalla, lógica, datos, configuración y dependencias. Aprende a leer el mapa y tus órdenes señalarán la ruta exacta, no una intuición.

QUÉ APRENDERÁS

- Orientarse en la estructura de carpetas de un proyecto generado y ubicar la pieza correcta a modificar.

Todo proyecto tiene la misma forma

Cambian los nombres, no la lógica. Un proyecto generado por IA casi siempre separa: lo que se **ve** (frontend), lo que se **procesa** (backend/lógica), lo que se **guarda** (datos), lo que se **ajusta** (configuración) y lo que se **pide prestado** (dependencias). Si sabes en qué cajón vive cada comportamiento, señalas la ruta correcta en vez de describir síntomas.

ACTIVIDAD INTERACTIVA DISPONIBLE EN
LUMBRE

1. Los cinco cajones del proyecto



Un mapa real, leído en voz alta

Cada cajón responde una pregunta distinta

```
mi-crm/\n|— public/          # lo que el navegador descarga (html, css, js)\n|
|— index.html\n|— src/\n|   |— routes/          # backend: QUÉ endpoints existen
|   |   (GET /contactos)\n|   |— services/      # lógica de negocio: CÓMO se valida y
|   |   calcula\n|   |— models/      # datos: QUÉ forma tienen contacto, empresa,
|   |   venta\n|   |— config/      # parámetros: a DÓNDE se conecta (base de datos,
|   |   APIs)\n|   |— .env          # SECRETOS: credenciales que NUNCA se suben al
|   |   repo\n|   |— package.json    # dependencias: QUÉ paquetes prestados se usan\n|   |— README.md        # el manual: para qué sirve y cómo arranca
```

Para reforzar

Dónde vive cada cambio

Nombrar el archivo en tu prompt no es tecnicismo: es impedir que la IA reintervenga lo estable.

Quieres...	Cajón correcto	No tocarías...
Cambiar un color del botón	frontend (public/src)	los models ni el .env
Agregar un campo «empresa» al contacto	datos (models) + backend (routes)	los estilos
Mover la app a otra base de datos	configuración (config / .env)	la lógica de services
Cambiar el mensaje de error al guardar	lógica (services)	las dependencias

Prueba

2. Quieres que el teléfono se guarde sin espacios ni guiones. ¿En qué cajón debe vivir el cambio?

- ☐ En public/, porque es lo que ve el usuario.
- ☐ En services/ (lógica) o models/ (regla del dato), según dónde validas.
- ☐ En .env, porque es un ajuste.
- ☐ En package.json.

3. Un archivo .env suele contener contraseñas y claves, por eso no debe subirse al repositorio ni compartirse con la IA sin limpiar.

- ☐ Verdadero
- ☐ Falso

Detecta el cambio fuera de lugar

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN
LUMBRE

4. Pediste «validar que el email tenga formato correcto». La IA tocó cuatro puntos. Señala el cambio mal ubicado que no aporta la validación:



5. Pide a la IA el árbol de carpetas de tu proyecto (solo nombres, no contenido). Asigna con una palabra qué hace cada carpeta. La próxima orden que des, señala el cajón exacto.

Un mapa claro en la cabeza se convierte en prompts que la IA no puede malinterpretar.

Referencia curricular

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

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



<https://empresa.lumbre.lat/object/fsi-m13-0a03>

Material educativo generado desde Lumbre · lumbre.lat