

L U M B R E



Arquitectura sin sintaxis: dibujar el sistema antes de pedirlo

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

Antes del primer prompt de código existe el plano: un dibujo con las piezas del sistema y cómo se conectan. No necesitas saber programar para dibujarlo; necesitas saber qué piezas hay. Aquí aprendes a representar la arquitectura — frontend, backend, datos, API— y a entregarle ese mapa a la IA para que construya sobre un plano y no sobre una intuición.

QUÉ APRENDERÁS

- Dibujar el mapa de componentes y conexiones de una aplicación y usarlo como guía en el prompt.

El plano antes del ladrillo



Entre piezas, como entre empresas, hay contratos: quién llama a quién, con qué datos y de quién es la responsabilidad. Un plano sin contrato es una postal.

Un constructor no empieza sin plano. Tú tampoco deberías empezar a pedir código sin saber **qué piezas** va a tener tu sistema y **cómo se hablan entre ellas**. Dibujar la arquitectura no es programar: es decidir de antemano el mapa. Cuando el mapa está claro, cada prompt deja de ser una apuesta.

ACTIVIDAD INTERACTIVA
DISPONIBLE EN LUMBRE

1. Las cuatro piezas de casi toda aplicación web



Qué es cada pieza (en una frase)

Nombra la pieza y su trabajo; la IA pone la sintaxis.

Pieza	Lo que hace	No necesitas saber...
Frontend	Lo que el usuario ve y toca en el navegador	el CSS exacto de cada botón
Backend	La lógica que decide qué pasa con lo que se envía	la sintaxis del servidor
Base de datos	Donde se guarda la información que persiste	optimizar índices
API	El puente por el que frontend y backend se hablan	firmar cada endpoint a mano

Cómo usar el plano en el prompt

Cuando le entregas el mapa a la IA, dejas de decir «hazme una app» y empiezas a decir «esto vive aquí, esto se conecta allá». El mismo plano se pega en cada prompt para que la IA no invente una arquitectura distinta en cada respuesta.

El plano como constraint

```
ARQUITECTURA DEL PROYECTO (respétala, no la cambies sin avisar):\n- Frontend:\n  pagina de agenda de contactos (lista + formulario).\n- API interna: GET\n  /contactos, POST /contactos, PUT /contactos/{id}, DELETE /contactos/{id}.\n- Backend: valida y guarda en la base de datos.\n- Base de datos: tabla\n  contactos(id, nombre, telefono, email, creado_en).\n\nConstruye SOLO el endpoint\nGET /contactos siguiendo este mapa. No toques el frontend todavía.
```

Para profundizar

Prueba

2. En tu plano, el formulario de registro del usuario guarda el usuario. ¿Qué pieza decide realmente si el correo es válido y lo escribe en la base de datos?

- ☐ El navegador del usuario.
- ☐ El backend (a través de la API).
- ☐ El DNS del dominio.
- ☐ La hoja de estilos CSS.

3. Dibujar la arquitectura exige saber escribir el código de cada componente.

- ☐ Verdadero
- ☐ Falso

Detecta el plano roto

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN
LUMBRE

4. Un alumno describe la arquitectura de su app. Señala la conexión que NO tiene sentido en un sistema web típico:



5. Dibuja (a mano o como lista) la arquitectura de tu proyecto: nombra cada pieza y qué le envía a cuál. Pégala al inicio de tu próximo prompt de código.

Si no sabes cómo se conecta una pieza con otra, ahí tienes un hueco de diseño que la IA rellenará por ti sin consultar.

Referencia curricular

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

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



<https://empresa.lumbre.lat/object/fsi-m05-0a01>

Material educativo generado desde Lumbre · lumbre.lat