

L U M B R E



CRUD: las cuatro operaciones que sostienen el 90 % de las aplicaciones

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

Agenda, CRM, tienda, gestor de tareas: casi todo sistema es, en el fondo, cuatro operaciones sobre datos repetidas para muchas entidades. Crear, Leer, Actualizar y Borrar —CRUD— es la rejilla con la que puedes describir cualquier función sin equívocos. Domínala y tus prompts de backend serán imposibles de malinterpretar.

QUÉ APRENDERÁS

- Reconocer crear, leer, actualizar y borrar en cualquier funcionalidad y pedirlos con precisión a la IA.

Cuatro verbos que lo cubren casi todo

Crear, **R**ead (leer), **U**ppdate (actualizar), **D**elete (borrar). Es el ciclo de vida de cualquier dato: nace, se consulta, se corrige, muere. En español también se llama **ABM** (Alta, Baja, Modificación). Cuando describes una pantalla como «gestión de clientes», lo que pides en realidad son estas cuatro operaciones sobre la entidad cliente.

Cada operación sobre una entidad se vuelve una ventanilla con nombre claro

- **CRUD**
 - Crear
 - POST /clientes
 - Leer
 - GET /clientes
 - GET /clientes/{id}
 - Actualizar
 - PUT /clientes/{id}
 - Borrar
 - DELETE /clientes/{id}

El mapa mental interactivo es interactivo en la versión online.



La rejilla CRUD sobre «Contactos»

Cinco filas que describen por completo una app de contactos. Añades entidades y repites la rejilla.

Operación	Lo que el usuario hace	Endpoint	Qué valida el servidor
Crear	Rellena el formulario y guarda	POST /contactos	nombre y teléfono obligatorios
Leer (lista)	Abre la agenda	GET /contactos	solo los del usuario dueño
Leer (uno)	Toca un contacto	GET /contactos/{id}	que el id exista
Actualizar	Edita y guarda cambios	PUT /contactos/{id}	que exista y los campos sean válidos
Borrar	Pulsa eliminar	DELETE /contactos/{id}	confirmación y propiedad del registro

Pide el CRUD completo, no a medias

Un solo prompt, ciclo completo

Construye el CRUD para la entidad Contacto (campos: id, nombre, telefono, email):\n- POST /contactos crea uno; requiere nombre; devuelve 400 si falta.\n- GET /contactos lista todos del usuario logueado.\n- GET /contactos/{id} devuelve uno o 404 si no existe.\n- PUT /contactos/{id} actualiza; 404 si no existe.\n- DELETE /contactos/{id} borra; 404 si no existe.\n\nRegla: un usuario solo puede ver y tocar SUS contactos. Aplica esa comprobacion en cada operacion.

Prueba

1. Tu app deja crear y ver contactos, pero al editar uno y guardar no cambia nada. ¿Qué operación del CRUD no está implementada (o no se pide)?

- ☐ Create (POST).
- ☐ Read (GET).
- ☐ Update (PUT).
- ☐ Delete (DELETE).

2. Una aplicación que «solo lista» datos no necesita CRUD.

- ☐ Verdadero
- ☐ Falso

Detecta el hueco en el CRUD

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN
LUMBRE

3. Revisas la lista de endpoints de «Tareas» que propuso una IA. Señala el hueco que romperá el ciclo de vida del dato:



4. Enumera las entidades de tu proyecto (contactos, pedidos, tareas...) y para cada una escribe sus cuatro endpoints CRUD. Será tu esqueleto de backend.

Si una entidad no tiene las cuatro, ya sabes qué función le faltará al final.

Referencia curricular

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

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



<https://empresa.lumbre.lat/object/fsi-m07-0a02>

Material educativo generado desde Lumbre · lumbre.lat