

## LUMBRE



# Proyecto 4 · Aplicación que vive de una API externa

**Materia:** Desarrollo de Software con IA

07/10/2026

## Antes de empezar

Tu app se vuelve dependiente: el servicio que consulta no es tuyo, se cae, cambia y cobra por uso. La misión: una aplicación que consulte una API pública real —clima con geocodificación, mapas con rutas, o cotizaciones— con la clave escondida en tu backend (nunca en el navegador), caché para no gastar la cuota en cada clic, y manejo honesto de los cinco estados: ok, lento, vacío, error y sin cuota. El proyecto que te vuelve creíble ante cualquier desarrollador cuando hables de integraciones.

### QUÉ APRENDERÁS

- Integrar un servicio real (clima, mapas o cotizaciones) con manejo de errores y cuotas.

## La misión

Elige UNA API pública con plan gratuito y construye su interfaz bonita y útil: **clima** para ciudades (con buscador geocodificado), **mapas** (punto a punto con distancia y tiempo), o **cotizaciones** (USD/MXN con gráfico histórico). Tres reglas de ingeniería que definen el proyecto: la clave del servicio vive en TU servidor (la señal 1 del detective aplicada a APIs), las respuestas se cachéan según su vida útil real (el clima de una hora no cambia cada segundo), y los cinco estados del servicio se muestran al usuario sin dramas ni pantallazos de error crudos.

## La ruta, seis tiempos

ACTIVIDAD INTERACTIVA DISPONIBLE EN LUMBRE

### 1. Ruta del proyecto 4



## El prompt del proxy con caché

### Los cinco estados escritos como criterios, el timeout incluido

```
Proyecto 4: integracion con la API de [SERVICIO]. Documentacion real: [pega
ENDPOINT, parametros\ny respuesta de ejemplo]. La clave va en variable de entorno
SERVICIO_KEY: jamas en el\nfrontend ni en el repo.\n\nImplementa en mi backend
(stack del maestro) el endpoint proxy /app/datos?ciudad=X:\n1. llama a la API real
con SERVICIO_KEY desde entorno\n2. CACHE: si tengo respuesta de esa ciudad con
menos de [TTL segun el dato], la devuelvo\n  de la cache sin llamar; si ya
vencio, llamo y actualizo. Reporta X-Cache: HIT/MISS.\n3. ESTADOS: respuesta
normal | sin datos (la API devolvio vacio: lo digo, NO invento)\n  | error de la
API (5xx: mensaje amable, reintento una vez) | SIN CUOTA (429 del\n  proveedor:
le digo al usuario que espere, y NO reintento en bucle)\n  | la API tardo
demasiado (timeout 5 s: cortamos y avisamos)\n4. el log registra ciudad, estado y
duracion. Nunca la clave, nunca el cuerpo completo.\n\nPruebas automatizadas por
cada estado (con la API simulada: mocks). Frontend despues:\nesta capa se aprueba
sola primero.
```

## Qué entrega este proyecto

*Este es el proyecto que los entrevistadores prueban: cualquiera hace el clic feliz. La prueba de corte, cuota y ciudad inexistente es lo que separa la demo del producto.*

| Entregable        | Barra alta de este proyecto                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------|
| ESPECIFICACION.md | el contrato de la API externa copiado de su documentación OFICIAL (con enlace), y los cinco estados definidos con su copy |
| HALLAZGOS.md      | el prompt de la IA que usó un parámetro que la API real no tiene (¿verificado contra la documentación o "plausible"?)     |
| PRUEBAS.md        | los cinco estados provocados uno por uno: vacío, error, cuota agotada, lento y normal                                     |
| App publicada     | funciona con la red cortada a mitad, sin filtrar la clave, y sin gastar la cuota en cada clic                             |

## Prueba

**2. Tu app muestra el clima y la IA puso la clave del proveedor en el JavaScript del frontend «porque así responde más directo». ¿Qué dos principios rompió?**

- ☐ El caché y el timeout.
- ☐ La clave pública es práctica estándar en apps web: está en muchos sitios.
- ☐ La clave viaja al navegador (cualquiera la extrae con F12 y la gasta o vende) y se expone en el repositorio si el código se sube: el proxy con caché existe precisamente para esto.
- ☐ El manejo del estado 429 y del estado vacío.

**3. La API real devolvió una lista vacía para una ciudad: el prompt correcto para la IA es «asegúrate de que nunca devuelva vacío, compléméntalo con la ciudad más cercana».**

- ☐ Verdadero
- ☐ Falso

**4. Paso uno y sin atajos: abre la documentación OFICIAL de tu API elegida y anota a mano los tres datos del contrato (endpoint, parámetros, límite del plan gratis) antes de hablar con la IA. Cuando ella genere el código, tu primera pasada de revisión es ctrl-f contra lo que copiaste, no contra su memoria.**

Integrar es confiar en un servicio que no controlas: la documentación leída y los estados probados son tu prima de seguro.

---

---

---

---

---

## Referencia curricular

|                 |                               |
|-----------------|-------------------------------|
| <b>Materia:</b> | Desarrollo de Software con IA |
|-----------------|-------------------------------|

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



<https://empresa.lumbre.lat/object/fsi-pj-0a04>

Material educativo generado desde Lumbre · [lumbre.lat](https://lumbre.lat)