

L U M B R E



Sesiones y permisos: quién es quién dentro de tu aplicación

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

Un servidor no tiene memoria: cada petición llega «suelta». Para saber quién la envía y qué puede hacer, existen la sesión y el token. Aquí entiendes cómo viaja la identidad del usuario y cómo se convierte en permisos que el backend comprueba en cada operación, sin escribir una línea de código de más.

QUÉ APRENDERÁS

- Describir cómo viaja la identidad de un usuario (sesión, token) y cómo se aplican permisos en el backend.

El servidor no recuerda: hay que decirle quién eres cada vez

El HTTP es amnésico: cada petición es un desconocido nuevo. Para que el backend sepa **quién** llama y **qué puede hacer**, la identidad viaja con cada petición dentro de una **sesión** o un **token**. Los **permisos** son las reglas que deciden, con esa identidad en mano, si la operación se autoriza o se rechaza.

ACTIVIDAD INTERACTIVA DISPONIBLE
EN LUMBRE

1. La identidad viaja con cada petición



Sesión, token y permiso (en una frase)

Los conceptos se reutilizan de APIs; aquí los aplicas a quién puede tocar qué dato.

Concepto	Qué es	Analogía
Sesión	El «estado de estar conectado» del usuario	tu pulsera de acceso en un evento
Token	La credencial que viaja en cada petición	el boleto con tu nombre que muestras en cada puerta
Permiso	La regla de qué puede hacer esa identidad	que con tu pulsera entras a zona VIP pero no a cocina
Rol	Un paquete de permisos para un tipo de usuario	admin, editor, invitado

Para reforzar

Prompt de permisos

Autorizar en el servidor, operación por operación

Implementa control de acceso en el CRUD de Contactos:\n- en cada petición, lee el token; si no hay token válido, responde 401 (no autorizado).\n- un usuario normal solo puede listar, crear, editar y borrar SUS propios contactos (comprueba el campo owner_id).\n- un usuario con rol «admin» puede ver todos.\n- si intenta tocar un contacto ajeno, responde 403, no lo devuelvas vacío disfrazado de éxito.\n\nLa comprobación va en el SERVIDOR en cada endpoint, no solo ocultando botones en el frontend.

Prueba

2. En tu app, «Ana» no ve el botón de borrar los contactos de «Luis». Pero Luis sabe la dirección del endpoint y llama a DELETE /contactos/{id} sin permiso. ¿Qué debe pasar?

- ☐ Como no veía el botón, la petición debe fallar en el navegador.
- ☐ El servidor debe rechazarla con 403, porque el permiso se comprueba en el backend.
- ☐ Debe borrar, porque el endpoint existe.
- ☐ Debe pedir el token otra vez.

3. El token de un usuario debe viajar dentro de cada petición para que el backend sepa quién llama.

- ☐ Verdadero
- ☐ Falso

Detecta el permiso de juguete

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN
LUMBRE

4. Revisas cómo una IA aplicó permisos en tu app. Señala la decisión que da una falsa sensación de seguridad:



5. Define 2 roles para tu proyecto (p. ej. usuario y admin) y escribe qué puede y qué no puede hacer cada uno sobre tus entidades. Pídele a la IA que lo aplique en el servidor.

Si un permiso se puede saltar llamando directo a la API, no es un permiso: es decoración.

Referencia curricular

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

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



<https://empresa.lumbre.lat/object/fsi-m07-0a03>

Material educativo generado desde Lumbre · lumbre.lat