

L U M B R E



Roles y permisos: admin no es un botón, es una matriz

Materia: Desarrollo de Software con IA

07/10/2026

Antes de empezar

«Que el admin pueda con todo» no es un diseño: es una invitación al desastre. Los permisos se piensan como una matriz: cada rol, frente a cada acción, un sí o un no deliberado. Aquí aprendes a construir esa matriz con la mínima autoridad necesaria y a entregársela a la IA para que cada endpoint la obedezca en el servidor.

QUÉ APRENDERÁS

- Construir una matriz rol-permiso-acción y exigírsela a la IA como criterio de aceptación.

El poder se diseña en una tabla, no en un botón

Cuando solo hay «usuario» y «admin», el admin puede con todo y cualquier error de uno es un desastre total. La forma profesional es una **matriz de roles y permisos**: filas = roles, columnas = acciones, y en cada casilla decides sí o no. Esa matriz, y no una sensación, es lo que le pides a la IA que implemente.

Una matriz para un gestor de tareas

Cada casilla es una decisión tuya. La IA implementa EXACTAMENTE esta tabla, ni una fila más de poder.

Rol	Ver tarea	Crear	Editar la suya	Editar la ajena	Borrar	Ver a todos
Invitado	Sí (asignadas)	No	No	No	No	No
Miembro	Sí (proyecto)	Sí	Sí	No	Solo la suya	No
Manager	Sí	Sí	Sí	Sí	Sí	Sí
Admin	Sí	Sí	Sí	Sí	Sí	Sí (y gestiona roles)

Tres principios que evitan el abuso de poder

La matriz hace visible dónde puede colarse un abuso

- **Principios**
 - Mínima autoridad
 - Cada rol, solo lo que necesita
 - En el servidor

- El permiso se comprueba en el backend
- Verificable
 - Cada casilla es un criterio de prueba

El mapa mental interactivo es interactivo en la versión online.



Prompt con la matriz

La matriz como criterio de aceptación

```
Implementa control por roles con esta matriz EXACTA (ningun rol obtiene permiso extra sin que lo diga):\n- Invitado: solo VER tareas asignadas.\n- Miembro: ver/crear/editar SUS tareas; borrar solo las suyas.\n- Manager: ademas editar/ver tareas ajenas y del proyecto.\n- Admin: todo lo anterior + gestionar roles.\n\nReglas:\n- la comprobacion se hace en el SERVIDOR en cada endpoint, leyendo el rol del token\n- si un endpoint pide permiso que el rol no tiene, responde 403\n- al final, generame la lista de endpoints y que rol necesita cada uno para que yo la audite.
```

Prueba

1. Un «Invitado» logra borrar una tarea llamando directo al endpoint DELETE aunque su rol no lo permite. ¿Qué falló en la implementación?

- ☐ La matriz tenía mal la columna del invitado.
- ☐ El permiso no se comprobó en el servidor: solo se ocultó el botón.
- ☐ Faltó una fila para el admin.
- ☐ El token caducó.

2. Dar al rol «admin» el mayor número posible de permisos es una buena práctica.

- ☐ Verdadero
- ☐ Falso

Detecta el diseño de permisos defectuoso

ACTIVIDAD
INTERACTIVA
DISPONIBLE EN
LUMBRE

3. Revisas una matriz de roles que propuso una IA. Señala la decisión que rompe el principio de mínima autoridad:



4. Dibuja la matriz de roles y permisos de tu proyecto (3-4 roles, tus acciones reales). Conviértela en criterio de aceptación y pide a la IA que la aplique en el servidor y te la devuelva para auditar.

Si no escribiste la matriz, la IA decidirá quién puede hacer qué... y no consultó contigo.

Referencia curricular

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

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



<https://empresa.lumbre.lat/object/fsi-m10-0a03>

Material educativo generado desde Lumbre · lumbre.lat