

Cómo funciona el scope en la práctica

Cuatro comportamientos explican casi todas las preguntas del tipo "¿por qué esta persona ve eso?".

Los roles se suman

Un empleado con `hub_manager` + `report_viewer` puede operar el hub **y** consultar informes. Ninguno de los dos roles restringe al otro, y el orden en que los asignes da igual.

El scope filtra los datos automáticamente

Tener acceso a un módulo no significa ver todos sus datos. El sistema filtra todo al scope del empleado.

Un `report_viewer` cuyo scope está limitado a la marca "Pizza Express" abre la misma pantalla de informes que los demás, pero solo ve las cifras de esa marca. No hay nada oculto ni en gris: simplemente esos datos no están en sus resultados.

“ Conviene explicarlo directamente a las personas. Que dos compañeros comparen el mismo informe y vean totales distintos es el comportamiento esperado, no un fallo.

El scope se aplica igual a todos los roles

A diferencia de sistemas antiguos, el scope **no** se configura por rol. Si un empleado puede ver las tiendas 1, 2 y 3, eso aplica a todo lo que haga: hub, informes, métricas, configuración de tienda, todo.

Es una simplificación deliberada: permite razonar sobre la visibilidad de un empleado mirando una sola configuración, en lugar de reconciliar varias.

Las excepciones individuales mandan sobre los roles

Puedes conceder o denegar un permiso concreto a un empleado concreto, opcionalmente con fecha de caducidad.

Un deny siempre gana, conceda lo que conceda su rol. Ver [Excepciones individuales de permisos](#).

Cómo encajarlo todo

Cuando alguien reporta que no puede hacer algo, comprueba en este orden:

1. **¿Tiene un rol que conceda esa acción?** → [Acceso por sección](#)
2. **¿Hay un deny sobre esa persona?** → [Excepciones individuales de permisos](#)
3. **¿Los datos están simplemente fuera de su scope?** → [Configurar el scope de un empleado](#)

La mayoría de los casos se resuelven en el paso 3, y hasta que lo compruebas parecen un permiso que falta.