



FASE II — AUTH CON RBAC

Aprendizaje-Express-ISS-11

Features Roles y Resources: Catálogo de Autorización (58 Recursos y Roles vacíos)

Los Dos Extremos del Permiso: Sujeto (Role) y Objeto (Resource)

1. ROL (Sujeto de Autorización)

- Entidad: features/auth/roles/ · tabla 'roles'.
- Identificador: Nombre único en MAYÚSCULAS ('ADMIN', 'SELLER').
- Función: Agrupar permisos y responsabilidades del sistema.
- REGLA DE ORO: Los roles nacen VACÍOS en el ISS-11. Crear 'ADMIN' no concede ningún acceso por sí solo.
- Estado: Maneja borrado lógico mediante 'status = active/inactive'.

2. RECURSO (Objeto de Autorización)

- Entidad: features/auth/resources/ · tabla 'resources'.
- Identificador: Tupla única (method, path) en patrón.
- Ruta en Patrón: '/api/productos/:id' cubre todos los identificadores concretos.
- Diferenciación Verbo: 'GET /api/clientes' y 'POST /api/clientes' son dos recursos distintos.
- Catálogo Semilla: 58 recursos cargados desde 'resource-catalog.ts'.

Feature Roles: Estructura en 4 Capas, DTOs y Seeder

Contrato DTO

- CreateRoleDto: 'name' obligatorio, 'description' opcional.
- UpdateRoleDto: Reemplazo completo (sin 'status').
- PatchRoleDto: Partial<UpdateRoleDto>.
- RoleResponseDto: Mapeador plano 'toRoleResponse'.
- Normalización: 'name' se convierte a MAYÚSCULAS en el modelo.

Cadena de Capas

- RolesRepository: 'findByName' para validaciones de unicidad.
- RolesService: Comprueba 'assertNameAvailable' y lanza 409 Conflict si ya existe.
- RolesController: Hereda de 'BaseController' e invoca 'this.run'.
- RolesRoutes: Registra endpoints bajo '/api/roles' con modalidad JWT + RBAC.

Seeder Canónico

- Roles Sembrados: 'ADMIN' (administrador) y 'SELLER' (vendedor).
- Idempotencia: Usa 'findOrCreate' por nombre.
- Reactivación: Si un rol existía inactivo, el seeder lo reactiva a 'active'.
- Swagger: Documenta 7 endpoints bajo etiqueta 'Roles' con 'bearerSecurity'.

Feature Resources: El Catálogo Semilla de 58 Recursos en Código

El Archivo Fuente Única: resource-catalog.ts

- Definición en Código: Declara 'RESOURCE_CATALOG' con exactamente 58 entradas deterministas.
- Estructura: Tupla 'CatalogResource { method, path, description, seller? }'.
- Derivación Automática: 'SELLER_RESOURCES' se filtra dinámicamente con 'seller: true' (7 recursos de operación).
- Exclusión de Sesión: Las rutas '/api/sesion/*' y '/api/sesiones/*' NO son recursos RBAC (son modalidades OPEN y JWT).
- Seeder Reconciliador: 'resources.seeder.ts' recorre el catálogo e inserta/reactiva en la tabla 'resources'.

Resumen del Catálogo

- **Total Recursos: 58**
- **Permisos ADMIN: 58 (100%)**
- **Permisos SELLER: 7 (Operación)**
- **Restricción UQ: (method, path)**
- **Modalidad: JWT + RBAC**

Desglose por Grupos del Catálogo Semilla de 58 Recursos

Grupo de Funcionalidad	Cant. Recursos	Concesión SELLER	Descripción / Alcance
Cientes (/api/clientes)	7	2 (GET / y GET /:id)	CRUD + Desactivación de clientes
Tipos de Producto (/api/tipos-producto)	7	0	CRUD + Desactivación de familias/categorías
Productos (/api/productos)	7	2 (GET / y GET /:id)	CRUD + Desactivación de catálogo de productos
Ventas (/api/ventas)	3	3 (GET, GET/:id, POST)	Consulta y registro de transacciones comerciales
Detalle de Ventas (/api/detalle-ventas)	1	0 (Sólo ADMIN)	Consulta de líneas de productos en ventas
Usuarios (/api/usuarios)	9	0	CRUD identidades, contraseña y permisos efectivos
Roles (/api/roles)	7	0	CRUD administrativo de roles del sistema
Recursos (/api/recursos)	7	0	CRUD de puntos de acceso protegibles
Asignaciones Usuario-Rol	5	0	Gestión de tabla pivote role_users
Concesiones Rol-Recurso	5	0	Gestión de tabla pivote resource_roles

Reglas de Dominio: Restricción Única UQ(method, path) y Error 409

Control de Duplicados en Resources

- Restricción en BD: UNIQUE INDEX sobre la combinación '(method, path)'.
- Normalización Prevención: Method a MAYÚSCULAS ('POST') y Path mediante 'normalizePath()'.
- Método 'findByOperation': Comprueba la existencia previa de la tupla antes de insertar en 'ResourcesService'.
- Respuesta HTTP 409: Evita errores 500 no controlados devolviendo 'AppError(409, "Resource with operation ... already exists")'.

Impacto del Borrado Lógico

- Endpoint: 'PATCH /api/recursos/:id/deactivate'.
- Mecanismo: Establece 'status = inactive' sin eliminar el registro físico.
- Efecto Inmediato: Deshabilita el punto de acceso para TODOS los roles en tiempo real.
- Seguridad 'Deny by Default': Ningún rol puede validar una concesión sobre un recurso inactivo (devuelve 403 Forbidden).
- Reconciliación: El seeder vuelve a activar el recurso si se requiere recuperar la operatividad.

Puntos de Extensión e Integración en la Aplicación Express

1. Agregador de Rutas (routes/index.ts)

Instancia e integra 'RolesRoutes' y 'ResourcesRoutes', registrando los endpoints '/api/roles' y '/api/recursos'.

2. Configuración App (config/index.ts)

Importa 'Role' y 'Resource' en el arranque para registrar los esquemas Sequelize antes de sincronizar la BD.

3. SeedersRunner (seeders/index.ts)

Siembra la seguridad PRIMERO: ejecuta 'seedRoles()' (2) y 'seedResources()' (58) antes de cualquier seeder de negocio.

4. Registry Swagger (swagger/index.ts)

Incorpora 'rolesSwagger' y 'resourcesSwagger' fusionando los 14 endpoints bajo el esquema 'bearerAuth'.

Criterios de Cierre GATE & Definición de Hecho (DoD - ISS-11)

Checklist de Verificación Obligatoria

- ✓ **Compilación TypeScript:** 'npx tsc --noEmit' finaliza con 0 errores.
- ✓ **Inserción de Catálogo:** 'npm run db:seed' crea exactamente 2 roles ('ADMIN', 'SELLER') y 58 recursos.
- ✓ **Verificación SQL:** 'SELECT COUNT(*) FROM resources' es igual a 58 y 'SELECT COUNT(*) FROM roles' es igual a 2.
- ✓ **Control de Duplicados:** Insertar un recurso con la misma tupla (method, path) devuelve HTTP 409 Conflict.
- ✓ **Independencia de Permisos:** Los roles nacen VACÍOS; la asignación de concesiones se delega al ISS-12.
- ✓ **Habilitación Futura:** Prepara los dos catálogos sobre los cuales se construirá la matriz RBAC en el ISS-12.