

## Controllers\Rol.controller.js

```
1 // Importamos el objeto 'db' que contiene los modelos de la base de datos
2 const db = require('../Models');
3
4 // Definimos el método 'lista' que se encargará de obtener todas las instancias de 'Rol'
5 exports.lista = (req, res, next) =>{
6   // Utilizamos el método 'findAll' de Sequelize para obtener todas las instancias de '
  Rol'
7   db.Rol.findAll()
8   .then(roles => {
9     // Si la operación es exitosa, enviamos las instancias obtenidas como respuesta en
  formato JSON
10     res.json(roles);
11   }).catch(err => {
12     next(err); // Pasamos el error al middleware de manejo de errores
13   });
14 }
15
16 // Definimos el método 'filtrar' que se encargará de obtener las instancias de 'Rol' que
  coincidan con un valor específico en un campo específico
17 exports.filtrar = (req, res, next) =>{
18   // Obtenemos el campo y el valor de los parámetros de la ruta de la solicitud HTTP
19   const campo = req.params.campo;
20   const valor = req.params.valor;
21   // Utilizamos el método 'findAll' de Sequelize para obtener las instancias de 'Rol'
  que tienen ese valor en ese campo
22   db.Rol.findAll({
23     where: {
24       [campo]: valor
25     }
26   })
27   .then(roles => {
28     // Si la operación es exitosa, enviamos las instancias obtenidas como respuesta en
  formato JSON
29     res.json(roles);
30   }).catch(err => {
31     next(err); // Pasamos el error al middleware de manejo de errores
32   });
33 }
34
35 // Definimos el método 'nuevo' que se encargará de crear una nueva instancia de 'Rol'
36 exports.nuevo = (req, res, next) =>{
37   // Verificamos que los datos necesarios estén presentes en el cuerpo de la solicitud
  HTTP
38   if(!req.body.nombre){
39     // Si faltan datos, enviamos un mensaje de error con un código de estado HTTP 400
40     res.status(400).send({
41       message: "Faltan datos"
42     });
43     return;
44   }
45   // Creamos un objeto con los datos de la nueva instancia
46   const rol = {
47     nombre: req.body.nombre,
48     id_plan: req.body.id_plan
49   }
50   // Utilizamos el método 'create' de Sequelize para crear la nueva instancia en la base
  de datos
51   db.Rol.create(rol)
52   .then(data => {
```

```

53 // Si la operación es exitosa, enviamos la nueva instancia como respuesta en
formato JSON
54 res.json(data);
55 }).catch(err => {
56   next(err); // Pasamos el error al middleware de manejo de errores
57 });
58 }
59
60 // Definimos el método 'actualizar' que se encargará de actualizar una instancia existente
de 'Rol'
61 exports.actualizar = (req, res, next) =>{
62   // Obtenemos el ID de los parámetros de la ruta de la solicitud HTTP
63   const id = req.params.id;
64   // Utilizamos el método 'update' de Sequelize para actualizar la instancia en la base
de datos
65   db.Rol.update(req.body,{
66     where: {id_rol: id}
67   })
68   .then(num => {
69     // Si la operación es exitosa y se actualizó una instancia, enviamos un mensaje de
éxito
70     if(num == 1){
71       res.send({
72         message: "rol actualizado"
73       });
74     }else{
75       // Si no se pudo actualizar la instancia, enviamos un mensaje de error
76       res.send({
77         message: "No se pudo actualizar el rol"
78       });
79     }
80   }).catch(err => {
81     next(err); // Pasamos el error al middleware de manejo de errores
82   });
83 }
84
85 // Definimos el método 'eliminar' que se encargará de eliminar una instancia existente de
'Rol'
86 exports.eliminar = (req, res, next) =>{
87   // Obtenemos el ID de los parámetros de la ruta de la solicitud HTTP
88   const id = req.params.id;
89   // Imprimimos un mensaje en la consola indicando que se intentará eliminar la
instancia con ese ID
90   console.log(`Eliminar rol con id: ${id}`);
91   // Utilizamos el método 'destroy' de Sequelize para eliminar la instancia en la base
de datos
92   db.Rol.destroy({
93     where: {id_rol: id}
94   })
95   .then(num => {
96     // Si la operación es exitosa y se eliminó una instancia, enviamos un mensaje de
éxito
97     if(num == 1){
98       res.send({
99         message: "rol eliminado"
100       });
101     }else{
102       // Si no se pudo eliminar la instancia, enviamos un mensaje de error
103       res.send({
104         message: "No se pudo eliminar el rol"
105       });
106     }

```

```
107 |     }).catch(err => {
108 |         next(err); // Pasamos el error al middleware de manejo de errores
109 |     });
110 | }
```