

## Controllers\Condicion.controller.js

```
1 // Importamos los modelos de la base de datos
2 const db = require('../Models');
3
4 // Definimos un controlador para obtener la lista de todas las condiciones
5 exports.lista = (req, res, next) =>{
6   // Utilizamos el método findAll de Sequelize para obtener todas las condiciones
7   db.Condicion.findAll()
8   .then(condiciones => {
9     // Si la operación es exitosa, enviamos las condiciones como respuesta en formato
10     JSON
11     res.json(condiciones);
12   }).catch(err => {
13     next(err); // Pasamos el error al middleware de manejo de errores
14   });
15 }
16
17 // Definimos un controlador para filtrar las condiciones por un campo y valor específicos
18 exports.filtrar = (req, res, next) =>{
19   // Obtenemos el campo y el valor de los parámetros de la ruta
20   const campo = req.params.campo;
21   const valor = req.params.valor;
22   // Utilizamos el método findAll de Sequelize para obtener las condiciones que cumplen
23   // con el filtro
24   db.Condicion.findAll({
25     where: {
26       [campo]: valor
27     }
28   })
29   .then(condiciones => {
30     // Si la operación es exitosa, enviamos las condiciones como respuesta en formato
31     JSON
32     res.json(condiciones);
33   }).catch(err => {
34     next(err); // Pasamos el error al middleware de manejo de errores
35   });
36 }
37
38 // Definimos un controlador para crear una nueva condición
39 exports.nuevo = (req, res, next) =>{
40   // Verificamos que los campos necesarios estén presentes en el cuerpo de la solicitud
41   if(!req.body.nombre){
42     // Si no están presentes, enviamos un mensaje de error con el estado 400
43     res.status(400).send({
44       message: "Faltan datos"
45     });
46     return;
47   }
48   // Creamos un objeto con los datos de la nueva condición
49   const condicion = {
50     nombre: req.body.nombre
51   }
52   // Utilizamos el método create de Sequelize para crear la nueva condición
53   db.Condicion.create(condicion)
54   .then(data => {
55     // Si la operación es exitosa, enviamos los datos de la nueva condición como
56     // respuesta en formato JSON
57     res.json(data);
58   }).catch(err => {
```

```
55     next(err); // Pasamos el error al middleware de manejo de errores
56   });
57 }
58
59 // Definimos un controlador para actualizar una condición existente
60 exports.actualizar = (req, res, next) => {
61   // Obtenemos el ID de la condición de los parámetros de la ruta
62   const id = req.params.id;
63
64   // Utilizamos el método update de Sequelize para actualizar la condición
65   db.Condicion.update(req.body, {
66     where: { id: id }
67   })
68   .then(num => {
69     if (num == 1) {
70       // Si la operación es exitosa, enviamos un mensaje de éxito
71       res.send({
72         message: "La condición fue actualizada exitosamente."
73       });
74     } else {
75       // Si no se actualizó ninguna condición, enviamos un mensaje de error
76       res.send({
77         message: `No se pudo actualizar la condición con ID=${id}.`
78       });
79     }
80   }).catch(err => {
81     next(err); // Pasamos el error al middleware de manejo de errores
82   });
83 };
84
85 // Definimos un controlador para eliminar una condición existente
86 exports.eliminar = (req, res, next) => {
87   // Obtenemos el ID de la condición de los parámetros de la ruta
88   const id = req.params.id;
89
90   // Utilizamos el método destroy de Sequelize para eliminar la condición
91   db.Condicion.destroy({
92     where: { id: id }
93   })
94   .then(num => {
95     if (num == 1) {
96       // Si la operación es exitosa, enviamos un mensaje de éxito
97       res.send({
98         message: "La condición fue eliminada exitosamente."
99       });
100     } else {
101       // Si no se eliminó ninguna condición, enviamos un mensaje de error
102       res.send({
103         message: `No se pudo eliminar la condición con ID=${id}.`
104       });
105     }
106   }).catch(err => {
107     next(err); // Pasamos el error al middleware de manejo de errores
108   });
109 };
```