

Controllers\Calificacion.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 calificaciones
5 exports.lista = (req, res, next) =>{
6   // Utilizamos el método findAll de Sequelize para obtener todas las calificaciones
7   db.Calificacion.findAll()
8   .then(calificaciones => {
9     // Si la operación es exitosa, enviamos las calificaciones como respuesta en
    formato JSON
10     res.json(calificaciones);
11   }).catch(err => {
12     next(err); // Pasamos el error al middleware de manejo de errores
13   });
14 }
15
16 // Definimos un controlador para filtrar las calificaciones por un campo y valor
    específicos
17 exports.filtrar = (req, res, next) =>{
18   // Obtenemos el campo y el valor de los parámetros de la ruta
19   const campo = req.params.campo;
20   const valor = req.params.valor;
21   // Utilizamos el método findAll de Sequelize para obtener las calificaciones que
    cumplen con el filtro
22   db.Calificacion.findAll({
23     where: {
24       [campo]: valor
25     }
26   })
27   .then(calificaciones => {
28     // Si la operación es exitosa, enviamos las calificaciones como respuesta en
    formato JSON
29     res.json(calificaciones);
30   }).catch(err => {
31     next(err); // Pasamos el error al middleware de manejo de errores
32   });
33 }
34
35 // Definimos un controlador para crear una nueva calificación
36 exports.nuevo = (req, res, next) =>{
37   // Verificamos que los campos necesarios estén presentes en el cuerpo de la solicitud
38   if(!req.body.calificacion || req.body.aprobado === undefined){
39     // Si no están presentes, enviamos un mensaje de error con el estado 400
40     res.status(400).send({
41       message: "Faltan datos"
42     });
43     return;
44   }
45   // Creamos un objeto con los datos de la nueva calificación
46   const calificacion = {
47     calificacion: req.body.calificacion,
48     aprobado: req.body.aprobado
49   }
50   // Utilizamos el método create de Sequelize para crear la nueva calificación
51   db.Calificacion.create(calificacion)
52   .then(data => {
53     // Si la operación es exitosa, enviamos los datos de la nueva calificación 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 un controlador para actualizar una calificación
61 exports.actualizar = (req, res, next) =>{
62   // Obtenemos el id de la calificación de los parámetros de la ruta
63   const id = req.params.id;
64   // Utilizamos el método update de Sequelize para actualizar la calificación
65   db.Calificacion.update(req.body,{
66     where: {id_calificacion: id}
67   })
68   .then(num => {
69     // Si la operación es exitosa y se actualizó una calificación, enviamos un mensaje
de éxito
70     if(num == 1){
71       res.send({
72         message: "calificacion actualizada"
73       });
74     }else{
75       // Si no se actualizó ninguna calificación, enviamos un mensaje de error
76       res.send({
77         message: "No se pudo actualizar la calificación"
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 calificación
86 exports.eliminar = (req, res, next) =>{
87   // Obtenemos el id de la calificación de los parámetros de la ruta
88   const id = req.params.id;
89   // Imprimimos un mensaje en la consola indicando que se va a eliminar la calificación
con el id especificado
90   console.log(`Eliminar calificación con id: ${id}`);
91   // Utilizamos el método destroy de Sequelize para eliminar la calificación
92   db.Calificacion.destroy({
93     where: {id_calificacion: id}
94   })
95   .then(num => {
96     // Si la operación es exitosa y se eliminó una calificación, enviamos un mensaje
de éxito
97     if(num == 1){
98       res.send({
99         message: "calificación eliminada"
100       });
101     }else{
102       // Si no se eliminó ninguna calificación, enviamos un mensaje de error
103       res.send({
104         message: "No se pudo eliminar la calificación"
105       });
106     }
107   }).catch(err => {
108     next(err); // Pasamos el error al middleware de manejo de errores
109   });
110 }
```