

Controllers\Division.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 divisiones
5 exports.lista = (req, res, next) =>{
6   // Utilizamos el método findAll de Sequelize para obtener todas las divisiones
7   db.Division.findAll()
8   .then(divisiones => {
9     // Si la operación es exitosa, enviamos las divisiones como respuesta en formato
10     JSON
11     res.json(divisiones);
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 divisiones 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 divisiones que cumplen
23   con el filtro
24   db.Division.findAll({
25     where: {
26       [campo]: valor
27     }
28   })
29   .then(divisiones => {
30     // Si la operación es exitosa, enviamos las divisiones como respuesta en formato
31     JSON
32     res.json(divisiones);
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 división
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 división
49   const division = {
50     nombre: req.body.nombre,
51     id_curso: req.body.id_curso
52   }
53   // Utilizamos el método create de Sequelize para crear la nueva división
54   db.Division.create(division)
55   .then(data => {
56     // Si la operación es exitosa, enviamos los datos de la nueva división como
57     respuesta en formato JSON
58     res.json(data);
59   });
60 }
```

```

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 división existente
61 exports.actualizar = (req, res, next) =>{
62     // Obtenemos el id de la división de los parámetros de la ruta
63     const id = req.params.id;
64     // Utilizamos el método update de Sequelize para actualizar la división
65     db.Division.update(req.body,{
66         where: {id_division: id}
67     })
68     .then(num => {
69         // Si la operación es exitosa y se actualizó una división, enviamos un mensaje de
éxito
70         if(num == 1){
71             res.send({
72                 message: "División actualizada"
73             });
74         }else{
75             // Si no se actualizó ninguna división, enviamos un mensaje de error
76             res.send({
77                 message: "No se pudo actualizar la división"
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 división existente
86 exports.eliminar = (req, res, next) =>{
87     // Obtenemos el id de la división de los parámetros de la ruta
88     const id = req.params.id;
89     // Utilizamos el método destroy de Sequelize para eliminar la división
90     db.Division.destroy({
91         where: {id_division: id}
92     })
93     .then(num => {
94         // Si la operación es exitosa y se eliminó una división, enviamos un mensaje de
éxito
95         if(num == 1){
96             res.send({
97                 message: "División eliminada"
98             });
99         }else{
100             // Si no se eliminó ninguna división, enviamos un mensaje de error
101             res.send({
102                 message: "No se pudo eliminar la división"
103             });
104         }
105     }).catch(err => {
106         next(err); // Pasamos el error al middleware de manejo de errores
107     });
108 }

```