

Controllers\Curso.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 todos los cursos
5 exports.lista = (req, res, next) =>{
6   // Utilizamos el método findAll de Sequelize para obtener todos los cursos
7   db.Curso.findAll()
8   .then(cursos => {
9     // Si la operación es exitosa, enviamos los cursos como respuesta en formato JSON
10     res.json(cursos);
11   }).catch(err => {
12     next(err); // Pasamos el error al middleware de manejo de errores
13   });
14 }
15
16 // Definimos un controlador para filtrar los cursos 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 los cursos que cumplen con
  el filtro
22   db.Curso.findAll({
23     where: {
24       [campo]: valor
25     }
26   })
27   .then(cursos => {
28     // Si la operación es exitosa, enviamos los cursos como respuesta en formato JSON
29     res.json(cursos);
30   }).catch(err => {
31     next(err); // Pasamos el error al middleware de manejo de errores
32   });
33 }
34
35 // Definimos un controlador para crear un nuevo curso
36 exports.nuevo = (req, res, next) =>{
37   // Verificamos que los campos necesarios estén presentes en el cuerpo de la solicitud
38   if(!req.body.nombre){
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 del nuevo curso
46   const curso = {
47     nombre: req.body.nombre,
48     id_plan: req.body.id_plan
49   }
50   // Utilizamos el método create de Sequelize para crear el nuevo curso
51   db.Curso.create(curso)
52   .then(data => {
53     // Si la operación es exitosa, enviamos los datos del nuevo curso 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 un curso
61 exports.actualizar = (req, res, next) =>{
62     // Obtenemos el ID del curso de los parámetros de la ruta
63     const id = req.params.id;
64     // Utilizamos el método update de Sequelize para actualizar el curso
65     db.Curso.update(req.body,{
66         where: {id_curso: id}
67     })
68     .then(num => {
69         // Si la operación es exitosa y se actualizó un curso, enviamos un mensaje de
        éxito
70         if(num == 1){
71             res.send({
72                 message: "curso actualizado"
73             });
74         } else {
75             // Si no se actualizó ningún curso, enviamos un mensaje de error
76             res.send({
77                 message: `No se pudo actualizar el curso con ID=${id}.`
78             });
79         }
80     }).catch(err => {
81         next(err); // Pasamos el error al middleware de manejo de errores
82     });
83 }
84
85 // Definimos un controlador para eliminar un curso
86 exports.eliminar = (req, res, next) => {
87     // Obtenemos el ID del curso de los parámetros de la ruta
88     const id = req.params.id;
89     // Utilizamos el método destroy de Sequelize para eliminar el curso
90     db.Curso.destroy({
91         where: { id: id }
92     })
93     .then(num => {
94         // Si la operación es exitosa y se eliminó un curso, enviamos un mensaje de éxito
95         if (num == 1) {
96             res.send({
97                 message: "El curso fue eliminado exitosamente."
98             });
99         } else {
100             // Si no se eliminó ningún curso, enviamos un mensaje de error
101             res.send({
102                 message: `No se pudo eliminar el curso con ID=${id}.`
103             });
104         }
105     }).catch(err => {
106         next(err); // Pasamos el error al middleware de manejo de errores
107     });
108 };

```