

Controllers\CicloLectivo.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 ciclos lectivos
5 exports.lista = (req, res, next) =>{
6   // Utilizamos el método findAll de Sequelize para obtener todos los ciclos lectivos
7   db.CicloLectivo.findAll()
8   .then(ciclolectivos => {
9     // Si la operación es exitosa, enviamos los ciclos lectivos como respuesta en
    formato JSON
10     res.json(ciclolectivos);
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 ciclos lectivos 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 ciclos lectivos que
    cumplen con el filtro
22   db.CicloLectivo.findAll({
23     where: {
24       [campo]: valor
25     }
26   })
27   .then(ciclolectivos => {
28     // Si la operación es exitosa, enviamos los ciclos lectivos como respuesta en
    formato JSON
29     res.json(ciclolectivos);
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 ciclo lectivo
36 exports.nuevo = (req, res, next) =>{
37   // Verificamos que los campos necesarios estén presentes en el cuerpo de la solicitud
38   if(!req.body.anio){
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 ciclo lectivo
46   const ciclolectivo = {
47     anio: req.body.anio
48   }
49   // Utilizamos el método create de Sequelize para crear el nuevo ciclo lectivo
50   db.CicloLectivo.create(ciclolectivo)
51   .then(data => {
52     // Si la operación es exitosa, enviamos los datos del nuevo ciclo lectivo como
    respuesta en formato JSON
53     res.json(data);
```

```
54     }).catch(err => {
55         next(err); // Pasamos el error al middleware de manejo de errores
56     });
57 }
58
59 // Definimos un controlador para actualizar un ciclo lectivo
60 exports.actualizar = (req, res, next) =>{
61     // Obtenemos el id del ciclo lectivo de los parámetros de la ruta
62     const id = req.params.id;
63     // Utilizamos el método update de Sequelize para actualizar el ciclo lectivo
64     db.CicloLectivo.update(req.body,{
65         where: {id_ciclo: id}
66     })
67     .then(num => {
68         // Si la operación es exitosa y se actualizó un ciclo lectivo, enviamos un mensaje
de éxito
69         if(num == 1){
70             res.send({
71                 message: "ciclo lectivo actualizado"
72             });
73         }else{
74             // Si no se actualizó ningún ciclo lectivo, enviamos un mensaje de error
75             res.send({
76                 message: "No se pudo actualizar el ciclo lectivo"
77             });
78         }
79     }).catch(err => {
80         next(err); // Pasamos el error al middleware de manejo de errores
81     });
82 }
83
84 // Definimos un controlador para eliminar un ciclo lectivo
85 exports.eliminar = (req, res, next) =>{
86     // Obtenemos el id del ciclo lectivo de los parámetros de la ruta
87     const id = req.params.id;
88     // Imprimimos un mensaje en la consola indicando que se va a eliminar el ciclo lectivo
con el id especificado
89     console.log(`Eliminar ciclo lectivo con id: ${id}`);
90     // Utilizamos el método destroy de Sequelize para eliminar el ciclo lectivo
91     db.CicloLectivo.destroy({
92         where: {id_ciclo: id}
93     })
94     .then(num => {
95         // Si la operación es exitosa y se eliminó un ciclo lectivo, enviamos un mensaje
de éxito
96         if(num == 1){
97             res.send({
98                 message: "ciclo lectivo eliminado"
99             });
100         }else{
101             // Si no se eliminó ningún ciclo lectivo, enviamos un mensaje de error
102             res.send({
103                 message: "No se pudo eliminar el ciclo lectivo"
104             });
105         }
106     }).catch(err => {
107         next(err); // Pasamos el error al middleware de manejo de errores
108     });
109 }
```