

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

```
56     }).catch(err => {
57         next(err); // Pasamos el error al middleware de manejo de errores
58     });
59 }
60
61 // Definimos un controlador para actualizar un alumno
62 exports.actualizar = (req, res, next) =>{
63     // Obtenemos el id del alumno de los parámetros de la ruta
64     const id = req.params.id;
65     // Utilizamos el método update de Sequelize para actualizar el alumno
66     db.Alumno.update(req.body,{
67         where: {dni: id}
68     })
69     .then(num => {
70         // Si la operación es exitosa y se actualizó un alumno, enviamos un mensaje de
71         éxito
72         if(num == 1){
73             res.send({
74                 message: "alumno actualizado"
75             });
76         }else{
77             // Si no se actualizó ningún alumno, enviamos un mensaje de error
78             res.send({
79                 message: "No se pudo actualizar el alumno"
80             });
81         }
82     }).catch(err => {
83         next(err); // Pasamos el error al middleware de manejo de errores
84     });
85 }
86
87 // Definimos un controlador para eliminar un alumno
88 exports.eliminar = (req, res, next) =>{
89     // Obtenemos el id del alumno de los parámetros de la ruta
90     const id = req.params.id;
91     // Utilizamos el método destroy de Sequelize para eliminar el alumno
92     db.Alumno.destroy({
93         where: {dni: id}
94     })
95     .then(num => {
96         // Si la operación es exitosa y se eliminó un alumno, enviamos un mensaje de éxito
97         if(num == 1){
98             res.send({
99                 message: "alumno eliminado"
100             });
101         }else{
102             // Si no se eliminó ningún alumno, enviamos un mensaje de error
103             res.send({
104                 message: "No se pudo eliminar el alumno"
105             });
106         }
107     }).catch(err => {
108         next(err); // Pasamos el error al middleware de manejo de errores
109     });
110 }
```