

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

```

54     // Si la operación es exitosa, enviamos los datos de la nueva fecha de examen 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 una fecha de examen existente
62 exports.actualizar = (req, res, next) =>{
63   // Obtenemos el id de la fecha de examen de los parámetros de la ruta
64   const id = req.params.id;
65   // Utilizamos el método update de Sequelize para actualizar la fecha de examen
66   db.FechaExamen.update(req.body,{
67     where: {id_fechaExamen: id}
68   })
69   .then(num => {
70     // Si la operación es exitosa y se actualizó una fecha de examen, enviamos un
    mensaje de éxito
71     if(num == 1){
72       res.send({
73         message: "fechaExamen actualizado"
74       });
75     }else{
76       // Si no se actualizó ninguna fecha de examen, enviamos un mensaje de error
77       res.send({
78         message: "No se pudo actualizar la fecha de examen"
79       });
80     }
81   }).catch(err => {
82     next(err); // Pasamos el error al middleware de manejo de errores
83   });
84 }
85
86 // Definimos un controlador para eliminar una fecha de examen existente
87 exports.eliminar = (req, res, next) =>{
88   // Obtenemos el id de la fecha de examen de los parámetros de la ruta
89   const id = req.params.id;
90   // Imprimimos un mensaje en la consola indicando que se va a eliminar una fecha de
    examen
91   console.log(`Eliminar fechaExamen con id: ${id}`);
92   // Utilizamos el método destroy de Sequelize para eliminar la fecha de examen
93   db.FechaExamen.destroy({
94     where: {id_fechaExamen: id}
95   })
96   .then(num => {
97     // Si la operación es exitosa y se eliminó una fecha de examen, enviamos un
    mensaje de éxito
98     if(num == 1){
99       res.send({
100         message: "fechaExamen eliminado"
101       });
102     }else{
103       // Si no se eliminó ninguna fecha de examen, enviamos un mensaje de error
104       res.send({
105         message: "No se pudo eliminar la fecha de examen"
106       });
107     }
108   }).catch(err => {
109     next(err); // Pasamos el error al middleware de manejo de errores
110   });

```

