

Controllers\Inscripcion.controller.js

```
1 // Importamos el modelo de la base de datos
2 const db = require('../Models');
3
4 // Función para obtener la lista de todas las inscripciones
5 exports.lista = (req, res, next) =>{
6   // Buscamos todas las inscripciones en la base de datos
7   db.Inscripcion.findAll()
8   .then(inscripciones => {
9     // Si la búsqueda es exitosa, devolvemos las inscripciones como respuesta en
    formato JSON
10     res.json(inscripciones);
11   }).catch(err => {
12     next(err); // Pasamos el error al middleware de manejo de errores
13   });
14 }
15
16 // Función para filtrar las inscripciones por un campo específico
17 exports.filtrar = (req, res, next) =>{
18   // Obtenemos el campo y el valor por el cual filtrar de los parámetros de la solicitud
19   const campo = req.params.campo;
20   const valor = req.params.valor;
21   // Buscamos todas las inscripciones que coincidan con el filtro
22   db.Inscripcion.findAll({
23     where: {
24       [campo]: valor
25     }
26   })
27   .then(inscripciones => {
28     // Si la búsqueda es exitosa, devolvemos las inscripciones filtradas como
    respuesta en formato JSON
29     res.json(inscripciones);
30   }).catch(err => {
31     next(err); // Pasamos el error al middleware de manejo de errores
32   });
33 }
34
35 // Función para crear una nueva inscripción
36 exports.nuevo = (req, res, next) =>{
37   // Verificamos que todos los campos necesarios estén presentes en el cuerpo de la
    solicitud
38   if(!req.body.id_previa || !req.body.id_turno || !req.body.id_fechaExamen ||
    !req.body.id_calificacion || !req.body.libro || !req.body.folio){
39     // Si falta algún campo, devolvemos un mensaje de error con un código de 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 inscripción
46   const inscripcion = {
47     id_previa: req.body.id_previa,
48     id_turno: req.body.id_turno,
49     id_fechaExamen: req.body.id_fechaExamen,
50     id_calificacion: req.body.id_calificacion,
51     libro: req.body.libro,
52     folio: req.body.folio
53   }
```

```

54 // Creamos la nueva inscripción en la base de datos
55 db.Inscripcion.create(inscripcion)
56 .then(data => {
57 // Si la creación es exitosa, devolvemos los datos de la nueva inscripción como
respuesta en formato JSON
58 res.json(data);
59 }).catch(err => {
60 next(err); // Pasamos el error al middleware de manejo de errores
61 });
62 }
63
64 // Función para actualizar una inscripción existente
65 exports.actualizar = (req, res, next) =>{
66 // Obtenemos el id de la inscripción a actualizar de los parámetros de la solicitud
67 const id = req.params.id;
68 // Actualizamos la inscripción en la base de datos con los datos presentes en el
cuerpo de la solicitud
69 db.Inscripcion.update(req.body,{
70 where: {id_inscripcion: id}
71 })
72 .then(num => {
73 // Si la actualización es exitosa, devolvemos un mensaje de éxito
74 if(num == 1){
75 res.send({
76 message: "inscripcion actualizado"
77 });
78 }else{
79 // Si no se pudo actualizar la inscripción, devolvemos un mensaje de error
80 res.send({
81 message: "No se pudo actualizar la inscripcion"
82 });
83 }
84 }).catch(err => {
85 next(err); // Pasamos el error al middleware de manejo de errores
86 });
87 }
88
89 // Función para eliminar una inscripción existente
90 exports.eliminar = (req, res, next) =>{
91 // Obtenemos el id de la inscripción a eliminar de los parámetros de la solicitud
92 const id = req.params.id;
93 // Imprimimos un mensaje en la consola indicando el id de la inscripción a eliminar
94 console.log(`Eliminar inscripcion con id: ${id}`);
95 // Eliminamos la inscripción de la base de datos
96 db.Inscripcion.destroy({
97 where: {id_inscripcion: id}
98 })
99 .then(num => {
100 // Si la eliminación es exitosa, devolvemos un mensaje de éxito
101 if(num == 1){
102 res.send({
103 message: "inscripcion eliminado"
104 });
105 }else{
106 // Si no se pudo eliminar la inscripción, devolvemos un mensaje de error
107 res.send({
108 message: "No se pudo eliminar la inscripcion"
109 });
110 }
111 }).catch(err => {

```

```
112 |         next(err); // Pasamos el error al middleware de manejo de errores
113 |     });
114 | }
```