Controllers\Previa.controller.js

```
// Importamos el objeto 'db' que contiene los modelos de la base de datos
const db = require('../Models');

// Definimos el método 'lista' que se encargará de obtener todas las instancias de 'Previa'
exports.lista = (req, res, next) =>{
    // Utilizamos el método 'findAll' de Sequelize para obtener todas las instancias de 'Previa'
    db.Previa.findAll()
    .then(previas => {
        // Si la operación es exitosa, enviamos las instancias obtenidas como respuesta en formato JSON
        res.json(previas);
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Definimos el método 'filtrar' que se encargará de obtener las instancias de 'Previa' que coincidan con un valor específico en un campo específico
exports.filtrar = (req, res, next) =>{
    // Obtenemos el campo y el valor de los parámetros de la ruta de la solicitud HTTP
    const campo = req.params.campo;
    const valor = req.params.valor;
    // Utilizamos el método 'findAll' de Sequelize para obtener las instancias de 'Previa' que tienen ese valor en ese campo
    db.Previa.findAll({
        where: {
            [campo]: valor
        }
    })
    .then(previas => {
        // Si la operación es exitosa, enviamos las instancias obtenidas como respuesta en formato JSON
        res.json(previas);
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Definimos el método 'nuevo' que se encargará de crear una nueva instancia de 'Previa'
exports.nuevo = (req, res, next) =>{
    // Verificamos que los datos necesarios estén presentes en el cuerpo de la solicitud HTTP
    if(!req.body.dni_alumno || !req.body.id_condicion || !req.body.id_materia){
        // Si faltan datos, enviamos un mensaje de error con un código de estado HTTP 400
        res.status(400).send({
            message: "Faltan datos"
        });
        return;
    }
    // Creamos un objeto con los datos de la nueva instancia
    const previa = {
        dni_alumno: req.body.dni_alumno,
        id_condicion: req.body.id_condicion,
        id_materia: req.body.id_materia,
        aprobado: req.body.aprobado ? req.body.aprobado : false
    }
```

```
    // Utilizamos el método 'create' de Sequelize para crear la nueva instancia en la base de datos
    db.Previa.create(previa)
    .then(data => {
        // Si la operación es exitosa, enviamos la nueva instancia como respuesta en formato JSON
        res.json(data);
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Definimos el método 'actualizar' que se encargará de actualizar una instancia existente de 'Previa'
exports.actualizar = (req, res, next) =>{
    // Obtenemos el ID de los parámetros de la ruta de la solicitud HTTP
    const id = req.params.id;
    // Utilizamos el método 'update' de Sequelize para actualizar la instancia en la base de datos
    db.Previa.update(req.body,{
        where: {id_previa: id}
    })
    .then(num => {
        // Si la operación es exitosa y se actualizó una instancia, enviamos un mensaje de éxito
        if(num == 1){
            res.send({
                message: "previa actualizado"
            });
        }else{
            // Si no se pudo actualizar la instancia, enviamos un mensaje de error
            res.send({
                message: "No se pudo actualizar la previa"
            });
        }
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Definimos el método 'eliminar' que se encargará de eliminar una instancia existente de 'Previa'
exports.eliminar = (req, res, next) =>{
    // Obtenemos el ID de los parámetros de la ruta de la solicitud HTTP
    const id = req.params.id;
    // Imprimimos un mensaje en la consola indicando que se intentará eliminar la instancia con ese ID
    console.log(`Eliminar previa con id: ${id}`);
    // Utilizamos el método 'destroy' de Sequelize para eliminar la instancia en la base de datos
    db.Previa.destroy({
        where: {id_previa: id}
    })
    .then(num => {
        // Si la operación es exitosa y se eliminó una instancia, enviamos un mensaje de éxito
        if(num == 1){
            res.send({
                message: "previa eliminado"
            });
        }else{
            // Si no se pudo eliminar la instancia, enviamos un mensaje de error
```

```
            res.send({
                message: "No se pudo eliminar la previa"
            });
        }
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}
```