Controllers\Materia.controller.js

```
// Importamos el modelo de la base de datos
const db = require('../Models');

// Función para obtener la lista de todas las materias
exports.lista = (req, res, next) =>{
    // Usamos el método findAll de Sequelize para obtener todas las materias
    db.Materia.findAll()
    .then(materias => {
        // Si la operación es exitosa, devolvemos las materias como respuesta JSON
        res.json(materias);
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Función para filtrar materias por un campo específico
exports.filtrar = (req, res, next) =>{
    // Obtenemos el campo y el valor de los parámetros de la solicitud
    const campo = req.params.campo;
    const valor = req.params.valor;
    // Buscamos todas las materias que coinciden con el valor en el campo especificado
    db.Materia.findAll({
        where: {
            [campo]: valor
        }
    })
    .then(materias => {
        // Si la operación es exitosa, devolvemos las materias como respuesta JSON
        res.json(materias);
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Función para crear una nueva materia
exports.nuevo = (req, res, next) =>{
    // Verificamos que los datos necesarios estén presentes en el cuerpo de la solicitud
    if(!req.body.nombre || !req.body.id_curso){
        // Si faltan datos, enviamos un mensaje de error con un código de estado 400
        res.status(400).send({
            message: "Faltan datos"
        });
        return;
    }
    // Creamos un objeto con los datos de la nueva materia
    const materia = {
        nombre: req.body.nombre,
        area: req.body.area,
        id_curso: req.body.id_curso
    }
    // Usamos el método create de Sequelize para crear la nueva materia
    db.Materia.create(materia)
    .then(data => {
        // Si la operación es exitosa, devolvemos los datos de la nueva materia como
respuesta JSON
        res.json(data);
    }).catch(err => {
```

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

// Función para actualizar una materia existente
exports.actualizar = (req, res, next) =>{
    // Obtenemos el ID de la materia de los parámetros de la solicitud
    const id = req.params.id;
    // Usamos el método update de Sequelize para actualizar la materia
    db.Materia.update(req.body,{
        where: {id_materia: id}
    })
    .then(num => {
        // Si la operación es exitosa, enviamos un mensaje de éxito
        if(num == 1){
            res.send({
                message: "Materia actualizada"
            });
        }else{
            // Si no se pudo actualizar la materia, enviamos un mensaje de error
            res.send({
                message: "No se pudo actualizar la materia"
            });
        }
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}

// Función para eliminar una materia existente
exports.eliminar = (req, res, next) =>{
    // Obtenemos el ID de la materia de los parámetros de la solicitud
    const id = req.params.id;
    // Imprimimos un mensaje en la consola indicando que vamos a eliminar la materia
    console.log(`Eliminar materia con id: ${id}`);
    // Usamos el método destroy de Sequelize para eliminar la materia
    db.Materia.destroy({
        where: {id_materia: id}
    })
    .then(num => {
        // Si la operación es exitosa, enviamos un mensaje de éxito
        if(num == 1){
            res.send({
                message: "Materia eliminada"
            });
        }else{
            // Si no se pudo eliminar la materia, enviamos un mensaje de error
            res.send({
                message: "No se pudo eliminar la materia"
            });
        }
    }).catch(err => {
        next(err); // Pasamos el error al middleware de manejo de errores
    });
}
```