



---

**El RA que trabajaremos y CE son:**

RA3.- Diseña, desarrolla y consume API RESTful aplicando buenas prácticas y estrategias de seguridad.

RA4.Gestiona datos y comunicación en tiempo real implementando estrategias de almacenamiento y seguridad.

## Sistema de Gestión de Academias y Cursos

Crear un sistema backend para gestionar varias academias, cursos, inscripciones y usuarios utilizan **Node.js**, **Express.js**, **MongoDB** y el **ODM Mongoose**

### Requisitos:

#### 1. Colecciones (definidas con Mongoose)

- **Usuarios**

```
{  
    "_id": "ObjectId",  
    "user": "nombre_usuario",  
    "email": "usuario@ejemplo.com",  
    "rol": "admin" | "user",  
    "password": "hashed_password"  
}
```

- **Academias**

```
{  
    "_id": "ObjectId",  
    "nombre": "nombre_academia",  
    "direccion": "direccion_academia",  
    "cursos": [ // Embedding  
        {  
            "cursoid": "ObjectId",  
            "titulo": "titulo_curso",  
            "precio": precio_curso  
        }  
    ]  
}
```

- **Cursos**

```
{  
    "_id": "ObjectId",  
    "titulo": "titulo_curso",  
    "descripcion": "descripcion_curso",  
    "precio": precio_curso,
```



```
        "categoryId": "ObjectId" // Referencing
    }
• Categorias
{
    "_id": "ObjectId",
    "nombre": "nombre_categoria",
    "descripcion": "descripcion_categoria"
}
```

## 2. Relaciones

- Cada academia tiene varios cursos disponibles (embedding).
- Cada curso pertenece a una categoría (referencing).
- Los modelos deben implementarse utilizando **ODM Mongoose**.

## 3. Pasos a seguir

- Configurar un servidor con **Express.js**.
- Conectar la aplicación a MongoDB usando **Mongoose**.
- Definir esquemas y modelos con Mongoose.
- Implementar rutas para la gestión de categorías, cursos y academias.
- Implementar registro de usuarios, login, generación de **JWT y refresh token**.
- Proteger las rutas sensibles mediante autenticación y autorización.
- Permitir solo a usuarios con rol **admin** crear, modificar o eliminar academias, cursos y categorías.
- Implementar verificación de cuenta por correo electrónico.

## 4. Rutas sugeridas

### 1. */api/categorias*

- Crear una categoría.
- Listar todas las categorías.
- Buscar categoría por nombre.
- Actualizar categoría.
- Eliminar categoría.

### 2. */api/cursos*

- Crear un curso.
- Listar todos los cursos.
- Listar cursos por categoría.
- Actualizar curso.
- Eliminar curso.



### 3. /api/academias

- Crear una academia.
- Listar todas las academias.
- Buscar academia por nombre.
- Actualizar academia.
- Eliminar academia.
- Agregar cursos a una academia.
- Listar cursos de una academia.
- Actualizar un curso embebido.
- Eliminar un curso embebido.

### 4. /api/usuarios

- Registrar usuario.
- Iniciar sesión.
- Refrescar token.

## 5. Consideraciones adicionales

- **Seguridad:** Hashear contraseñas antes de guardarlas.
- **Roles:** Solo administradores pueden modificar datos críticos.
- **Validaciones:** Usar validaciones con Mongoose en POST y PUT.
- **Manejo de errores:** Implementar respuestas claras y controladas.
- **Autenticación:** Middleware para proteger rutas privadas.