

## Implementación de un CRUD básico:

- Cree una aplicación Node.js que permita crear, leer, actualizar y eliminar (CRUD) registros en una base de datos (MySQL, SQLServer) a través de una API RESTful.
- Deberá asegurarse de que la **aplicación maneje correctamente los errores** y responda con los códigos de estado HTTP correctos.
- Deberá diseñar la base de datos con un mínimo de 3 tablas a las cuales consultar.
  - T1: Estudiante
  - T2: Curso
  - T3: Curso-estudiante
- Los datos por obtener son:
  1. Lista de cursos
  2. Lista de estudiantes
  3. Lista de estudiantes **con sus cursos asignados**
- Se consumirá el servicio por medio de **Postman**.

### 1. CREAR:

```
{
  ...."idEstudiante": "<ID>",
  ...."nombre": "<Nombre del estudiante>",
  ...."apellido": "<Apellidos del estudiante>",
  ...."nivel": "<Nivel>", //p.e.: Diversificado
  ...."seccion": "<sección>"
}

{
  ...."idCurso": "<código del curso>",
  ...."nombreCurso": "<nombre del curso>",
  ...."idGrado": "<código grado>", //p.e.: 4TO
  ...."idCarrera": "<carrera>", //p.e.: Bachillerato
  ...."catedratico": [
    ....{
      ...."idCatedratico": "<código del catedrático>", //p.e.: OGARCIA
      ...."nombreCatedratico": "<Nombre del catedrático>"
    }
  ]
}
```

## 2. LEER:

```
{
  .... "idEstudiante": "<ID>",
  .... "nombre": "<Nombre del estudiante>",
  .... "apellido": "<Apellidos del estudiante>",
  .... "nivel": "<Nivel>", //p.e.: Diversificado
  .... "grado": "<Grado>", //p.e.: 4to
  .... "idCarrera": "<código de carrera>", //p.e.: Bachillerato
  .... "seccion": "<sección>",
  .... "cursosAsignados": [
    .... {
    ....   "idCurso": "<código del curso>",
    ....   "nombreCurso": "<nombre del curso>"
    .... }
  .... ]
}
```

## 3. ACTUALIZAR:

```
{
  .... "idEstudiante": "<ID>",
  .... "nombre": "<Nuevo nombre de estudiante>"
}
```

## 4. ELIMINAR:

```
{
  .... "idEstudiante": "<ID>"
}
```

2. Implementación de un middleware de autenticación básico: cree un middleware de autenticación básico utilizando Node.js y Express. El middleware debe verificar si el usuario ha iniciado sesión antes de permitir que acceda a una ruta protegida. Utilice JWT (JSON Web Tokens) para manejar la autenticación.

## Condiciones:

- Los datos de acceso son a discreción del postulante.
- Subir a **Github** el código para revisión.