



## Ciclo 2

### Actividad: Parcial Práctico No. 2

1. Lea completamente las instrucciones antes de iniciar
2. Cree un proyecto nuevo en Nest.js
3. Suba el proyecto a un repositorio en su cuenta personal en GitHub

**MUY IMPORTANTE:** Durante el desarrollo del parcial, no olvide hacer  y  en su repositorio periódicamente.

Usted ha sido contratado por la decanatura de investigaciones para desarrollar un sistema de gestión de iniciativas lideradas por estudiantes de pregrado. En este sistema interactúan estudiantes, profesores y la decanatura de investigación. Su tarea es crear el API web que ayudará a resolver esta necesidad.

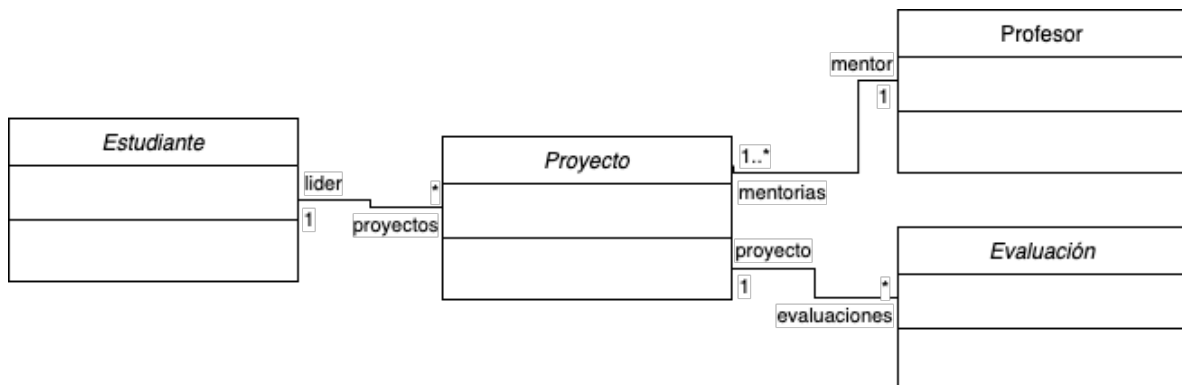


Ilustración 1. Diagrama de clases



## Ciclo 2

### Actividad: Parcial Práctico No. 2

## PARTE 1. Trabajo en clase (60%)

### Punto 1. Persistencia (25%)

(4%) Cree la entidad `Estudiante` la cual tiene un número de cedula (int), un nombre (string), semestre (int), un programa (string), un promedio (int), y un id (Long-Autogenerado).

(4%) Cree la entidad `Profesor` la cual tiene un cedula (int), nombre (string), departamento (string), extensión (int), esParEvaluado (boolean), y un id (Long-Autogenerado).

(4%) Cree la entidad `Proyecto` la cual tiene un título (string), un área (string), un presupuesto (int), una nota final (int), un estado (int), una fecha de inicio (string), una fecha de fin (string) y un id (Long-Autogenerado).

(3%) Cree la entidad `Asesoramiento` la cual tiene un id (Long-Autogenerado).

(10%) Cree todas las relaciones del diagrama UML.

### Punto 2. Lógica (25%)

(6%) Cree la clase correspondiente para la lógica de un `Estudiante`. **Implemente:**

- `crearEstudiante()` – Solo permitir si el promedio es mayor a 3.2 y semestre  $\geq 4$
- `eliminarEstudiante(id)`, No se puede eliminar si tiene proyectos activos.

(6%) Cree la clase correspondiente para la lógica de un `Profesor`. **Implemente:**

- `crearProfesor()` – Validar extensión con exactamente 5 dígitos,
- `asignarEvaluador()` - Solo si el profesor tiene menos de 3 evaluaciones activas,

(10%) Cree la clase correspondiente para la lógica de una `Proyecto`. **Implemente:**

- `crearProyecto ()` - Valide que el presupuesto  $> 0$ ,  $|\text{titulo}| > 15$ .
- `avanzarProyecto(id)`, - Un proyecto puede tener un estado entre 0 y 4, se debe sumar 1 en cada llamado y manejar el caso si un proyecto ya esta en su máximo estado.
- `findAllEstudiantes()` – Retorna el listado de todos los estudiantes relacionados al proyecto.

(3%) Cree la clase correspondiente para la lógica de una `Asesoramiento`. **Implemente:**

- `crearEvaluación()` - Validar que evaluador  $\neq$  mentor, calificación entre 0 y 5.



## Ciclo 2

### Actividad: Parcial Práctico No. 2

#### Punto 3. BONO Controlador (primera parte) (5%)

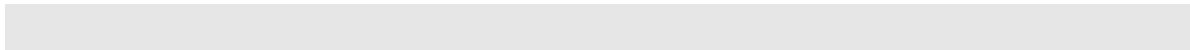
(5%) Implemente el controlador que permita usar el método `crearEvaluacion()`

#### Punto 4. Documentación API (10%)

(10%) Entregue en la hoja blanca, una definición de todos los endpoints del parcial (clase + casa) siguiendo el formato:

<metodo> <URI> <representacion> - <nombreDTOentrada> - <nombreDTosalida>

Defina brevemente en la parte de atrás de la hoja la información que compone cada DTO



Cuando finalice el ejercicio suba los cambios a su repositorio de GitHub.

Su repositorio debe contar con por lo menos 3 commits antes de la generación del release.

Haga un        en su repositorio con la etiqueta v1.0.0 y el título **parcial\_2\_clase**.

Suba el enlace del        como respuesta a la actividad **Parcial2Seccion2** de Bloque Neón.

Tenga en cuenta que no se revisaran cambios en las capa de persistencia y lógica después de este       .



## Ciclo 2

### Actividad: Parcial Práctico No. 2

#### PARTE 2. Trabajo en casa (40%)

##### Punto 1. Controladores (segunda parte) (15%)

(15%) Realice la implementación de los controladores NEST para exponer con una API REST los métodos faltantes generados en la capa de lógica.

##### Punto 2. Prueba de lógica (15%)

(15%) Implemente pruebas para los métodos desarrollados en la capa de lógica. Agregue por lo menos un caso positivo y un caso negativo.

##### Punto 3. Documentación POSTMAN (5%)

(5%) Cree la documentación en POSTMAN para el API REST creado. Debe incluir ejemplos de peticiones y resultados con códigos de respuesta.

##### Punto 4. Pruebas POSTMAN (5%)

(5%) Cree las pruebas en POSTMAN de los servicios creados por usted. Agregue una descripción de porque decidió probar los casos seleccionados.

Cuando finalice el ejercicio suba los cambios a su repositorio de GitHub

Haga un [commit](#) en su repositorio con la etiqueta v2.0.0 y el título **parcial\_2\_casa**.

Este release debe generarse antes del día domingo 18 de mayo de 2025.

Luego de finalizada la actividad no realice ningún cambio al repositorio. Cualquier modificación, por pequeña que sea, anula automáticamente el parcial.