



Ciclo 2

Actividad: Parcial Práctico No. 2

Instrucciones generales

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 *commit* y *push* en su repositorio periódicamente.

Enunciado

Usted ha sido contratado por el departamento de ingeniería de sistemas para diseñar el sistema de gestión de proyectos de grado del departamento. En este sistema interactúan profesores, estudiantes, los coordinadores académicos y el personal de la biblioteca. Se requiere implementar el API web que nos ayudara a manejar este sistema. El siguiente modelo de clases muestra las entidades y relaciones que usted debe crear en este examen.

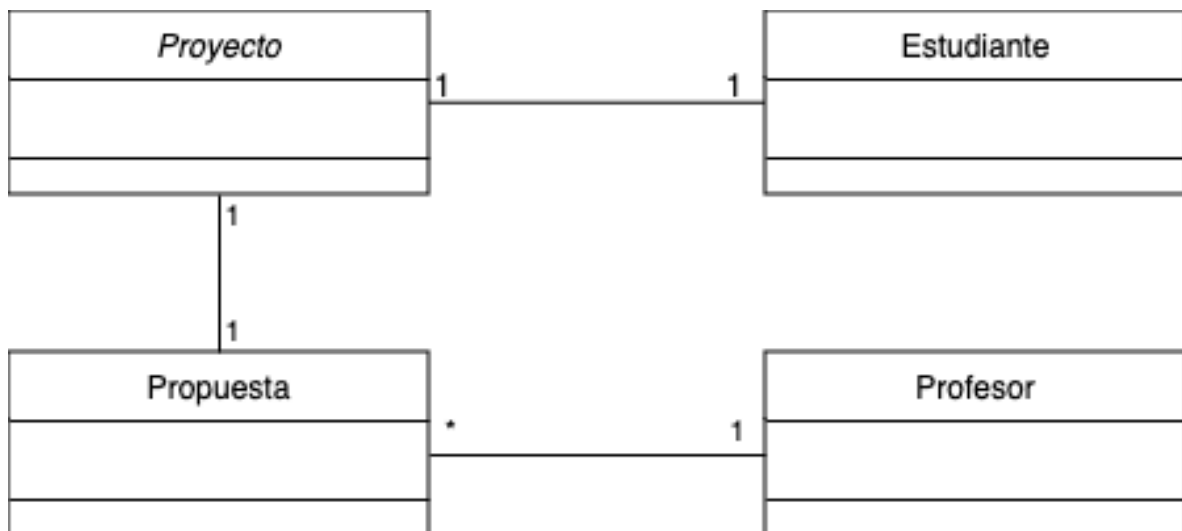


Ilustración 1. Diagrama de clases



Ciclo 2

Actividad: Parcial Práctico No. 2

PARTE 1. Trabajo en clase (60%)

Punto 1. Persistencia (25%)

(6%) Cree la entidad *Profesor* la cual tiene un número de cedula (int), un nombre (string), un grupo de investigación (string), un número de extensión (int), un grupo de Investigación (string) y un id (Long-Autogenerado).

(4%) Cree la entidad *Estudiante* la cual tiene un nombre (String), un código de estudiante (String), un número de créditos aprobados (int) y un id (Long-Autogenerado).

(4%) Cree la entidad *Propuesta* la cual tiene un título (String), una descripción (String), una palabra clave (String) y un id (Long-Autogenerado).

(4%) Cree la entidad *Proyecto* la cual tiene un fechaInicio (Date), una fechaFin (Date), una URL (String) y un id (Long-Autogenerado).

(7%) Cree todas las relaciones marcadas en rojo en el diagrama UML.

Punto 2. Lógica (25%)

(8%) Cree la clase correspondiente para la lógica de un *Profesor*. **Implemente:**

- **crearProfesor()** – se debe validar que el grupo de Investigación este entre los siguientes valores: TICSW, IMAGINE, COMIT,
- **findProfesorById(id)**,
- **eliminarProfesor(id)**, Un profesor no se puede eliminar si tiene una propuesta que tiene un proyecto asociado.
- **eliminarProfesor(cedula)** Un profesor no se puede eliminar si tiene una propuesta que tiene un proyecto asociado.

(7%) Cree la clase correspondiente para la lógica de una *Propuesta*. **Implemente:**

- **crearPropuesta()** – se debe validar que el título no sea vacío,
- **findPropuestaById(id)**,
- **findAllPropuesta()**,
- **deletePropuesta(id)** – Una propuesta no se puede eliminar si tiene un proyecto.



Ciclo 2

Actividad: Parcial Práctico No. 2

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

- **crearEstudiante()** - Valide que el código de estudiante tenga 10 caracteres,
- **findEstudianteById(id)**,

(4%) Cree la clase correspondiente para la lógica de un Proyecto. **Implemente:**

- **crearProyecto()**. Valide que la fecha de fin es posterior a la fecha de inicio.

Punto 3. Controladores (primera parte) (10%)

(10%) Implemente el controlador que permita retornar todas las propuestas con su información y relaciones.

Entrega trabajo en clase

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 [release](#) en su repositorio con la etiqueta v1.0.0 y el título **parcial_2_clase**.

Suba el enlace del *release* 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 *release*.



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.

Entrega

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

Haga un [release](#) en su repositorio con la etiqueta v2.0.0 y el título **parcial_2_casa**.

Este release debe generarse antes del día domingo 26 de mayo de 2024.

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