



## 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

La Universidad quiere promover la participación en eventos culturales como conciertos, obras de teatro, exposiciones y talleres. Para esto, desea una aplicación que permita a estudiantes registrarse en actividades, y a la Decanatura Cultural crear y gestionar dichas actividades. Además, cada evento puede tener una o más reseñas de los asistentes.

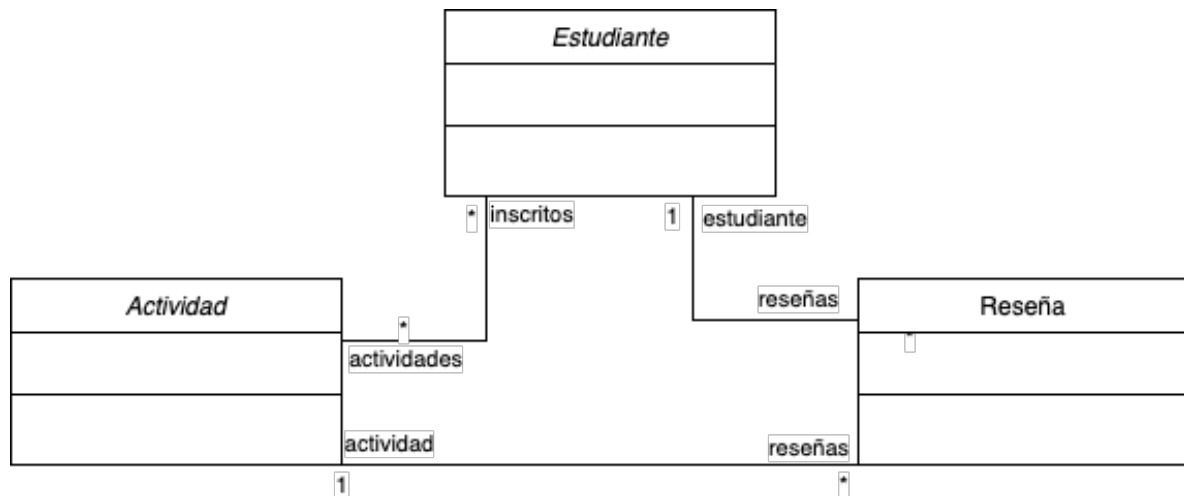


Ilustración 1. Diagrama de clases



## Ciclo 2

### Actividad: Parcial Práctico No. 2

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

Lea por completo el enunciado. Entregue a las 8:25 en la hoja blanca la respuesta al punto 4.

### Punto 1. Persistencia (15%)

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

(4%) Cree la entidad *Actividad* la cual tiene un título (string), un fecha (string), un cupoMaximo (int), estado (int), y un id (Long-Autogenerado).

(5%) Cree la entidad *Reseña* la cual tiene un comentario (string), una calificación (int), una fecha (string) y un id (Long-Autogenerado).

(7%) Cree todas las relaciones del diagrama UML.

### Punto 2. Lógica (30%)

(10%) Cree la clase correspondiente para la lógica de un *Estudiante*. **Implemente:**

- **crearEstudiante()** – email debe ser válido y semestre debe estar entre 1 y 10
- **findEstudianteById(id)**,
- **InscribirseActividad(estudianteID, actividadID)**, La actividad debe contar con cupo, debe estar en estado 0 y no debe existir una inscripción previa por parte del estudiante.

(10%) Cree la clase correspondiente para la lógica de un *Actividad*. **Implemente:**

- **crearActividad()** – se debe validar que el título tenga una longitud mínima de 15 caracteres y no puede tener simbolos,
- **cambiarEstado(actividadID, estado)** – una actividad tiene 3 estados posibles, 0 (abierta), 1(Cerrada), 2(Finalizada). Toda actividad se crea como abierta, solo puede ser cerrada si el 80% del cupo está lleno, o finalizada si no hay cupo,
- **findAllActividadesByDate(fecha: string)** – retorna todas las actividades con fecha idéntica a la pasada por parametro ,

(5%) Cree la clase correspondiente para la lógica de una *Reseña*. **Implemente:**

- **agregarReseña()** – la actividad está finalizada y el estudiante estuvo inscrito.



## Ciclo 2

### Actividad: Parcial Práctico No. 2

- `findClaseById(id)`,

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

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

### Punto 4. Documentación API (15%)

(15%) Entregue en la hoja blanca, una definición de todos los endpoints del parcial (clase + casa) siguiendo el formato:

<metodo> <URIparcial> - <nombreDTOentrada> - <nombreDTosalida>

Ej. PUT /recurso/:argumento – EntradaDTO - SalidaDTO

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

### 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.