



## 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 área de viajes de la Universidad para crear la aplicación que maneja los bonos de viaje para los profesores, estos bonos se les entrega a los profesores de acuerdo a su calificación en las encuestas de cursos. En este sistema interactúan profesores y decanatura. 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.

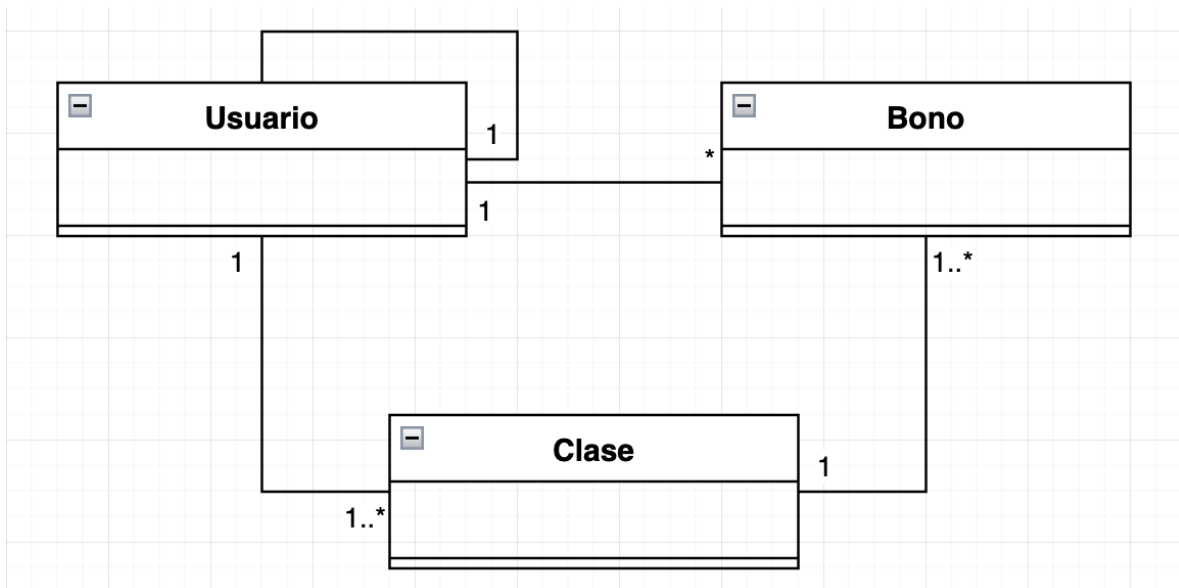


Ilustración 1. Diagrama de clases



## Ciclo 2

### Actividad: Parcial Práctico No. 2

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

### Punto 1. Persistencia (21%)

(6%) Cree la entidad *Usuario* la cual tiene un número de cedula (int), un nombre (string), un grupo de investigación (string), un número de extensión (int), un rol (String[Profesor, Decana]), un jefe (Usuario) y un id (Long-Autogenerado).

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

(4%) Cree la entidad *Bono* la cual tiene un monto (int), una calificación (double), una palabra clave (String) y un id (Long-Autogenerado).

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

### Punto 2. Lógica (25%)

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

- **crearUsuario()** – Si el rol es **Profesor** se debe validar que el grupo de Investigación este entre los siguientes valores: TICSW, IMAGINE, COMIT. Si el rol es **Decana** se debe validar que el número de extensión sea de 8 dígitos.
- **findUsuarioById(id)**,
- **eliminarUsuario(id)**, Un usuario no se puede eliminar si tiene rol de Decana o si tiene un bono asociado.

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

- **crearBono()** – se debe validar que el monto no sea vacío, positivo y el usuario debe tener rol de profesor,
- **findBonoByCodigo(cod)**,
- **findAllBonosByUsuario(userID)**,
- **deleteBono(id)** – Un bono no se puede eliminar si la calificación es mayor a 4.

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

- **crearClase()** - Valide que el código tenga 10 caracteres,
- **findClaseById(id)**,



## Ciclo 2

### Actividad: Parcial Práctico No. 2

#### Punto 3. Controladores (primera parte) (14%)

(14%) Implemente el controlador que permita usar todos los métodos del servicio de Bono.

#### 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.