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 una fábrica de software a cargo de la creación de las nuevas redes sociales (KitKot, YourSpace y HeadSpace), todas estas enfocadas en el uso de imágenes como principal método de interacción con el usuario. Se requiere implementar el API web que nos ayudara a manejar todas las redes sociales. Se tiene el siguiente modelo de clases, el API web debe seguir la estructura de directorios propuesta por clean-architecture (controllers, services, model).

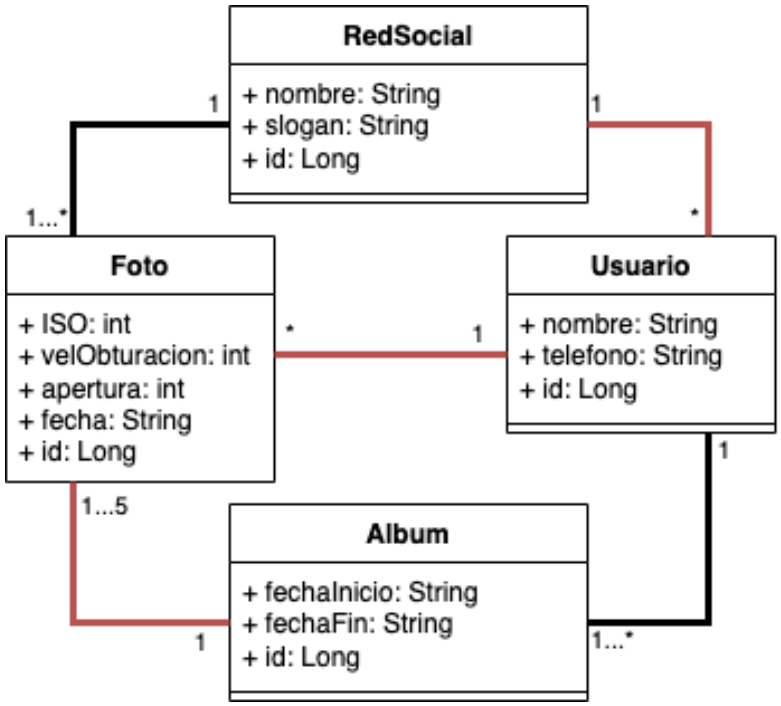

Ilustración 1. Diagrama de clases

Ciclo 2

Actividad: Parcial Práctico No. 2

PARTE 1. Trabajo en clase (70%)

Punto 1. Persistencia (30%)

(8%) Cree la entidad *FotoEntity* la cual tiene un ISO (int), una velObturacion (int), una apertura (Int), una fecha (String) y un id (Long-Autogenerado).

(6%) Cree la entidad *UsuarioEntity* la cual tiene un nombre (String), un teléfono (String) y un id (Long).

(6%) Cree la entidad *RedSocialEntity* la cual tiene un nombre (String), un slogan (String), y un id (Long).

(6%) Cree la entidad *Álbum* la cual tiene una fechaInicio (Date), una fechaFin (Date), un título (String) y un id (Long).

(4%) Cree todas las relaciones marcadas en rojo en el diagrama UML.

Punto 2. Lógica (30%)

(7%) Cree la clase correspondiente para la lógica del Álbum. **Implemente:**

- **createAlbum()** – se debe validar que el titúlo no sea vacío**,**
- **findAlbumById(id),**
- **addPhotoToAlbum() –** Se debe validar que la fecha de la foto esta entre las fechas de inicio y fin del album**,**
- **deleteAlbum(id) –** Un album no se puede eliminar si tiene alguna foto asignada**.**

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

- **createUsuario() -** Valide que el teléfono de un usuario tenga solo 10 caracteres**,**
- **findUsuarioById(id),**
- **findAllUsuarios()**.

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

- **createLibreria()**. Valide que slogan no está vacío y que tiene por lo menos 20 caracteres.

Ciclo 2
Actividad: Parcial Práctico No. 2

(11%) Cree la clase correspondiente para la lógica de la Foto. **Implemente:**

- **createFoto() –** Para crear un foto, se debe validar varias cosas:
  - **ISO:** su valor esta entre 100 y 6400
  - **valObturación:** su valor esta entre 2 y 250,
  - **apertura:** su valor debe estar entre 1 y 32
  - Al momento de crear una foto, máx 2 de estos valores deben estar por encima del valor medio de sus cotas.
- **findFotoByID(),**
- **findAllFotos(),**
- **deleteFoto() –** Si la foto es la última de un álbum, se debe eliminar el álbum también.

Punto 3. Prueba de lógica (primera parte) (10%)

(10%) Implemente las pruebas para los métodos createFoto(), deleteFoto() y deleteAlbum(). Debe probar 1 caso exitoso y un caso fallido.

Entrega trabajo en clase

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

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 ***Parcial2-Parte1*** 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 (30%)

Punto 3. Prueba de lógica (segunda parte) (10%)

(7%) Implemente pruebas para los métodos desarrollados en la capa de lógica que no se han probado hasta el momento. Agregue por lo menos un caso positivo y un caso negativo.

Punto 4. Controladores (7%)

(5%) Realice la implementación de los controladores NEST para exponer con una API REST los métodos generados en la capa de lógica.

Punto 5. Documentación POSTMAN (6%)

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

Punto 6. Pruebas POSTMAN (7%)

(7%) 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**.

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