



Ciclo 1 – Iteración 2

Actividad: Parcial Práctico No. 1

Enunciado

Se requiere realizar el API web para un *marketplace* de vinilos que tiene el modelo de clases descrito en la Ilustración 1.

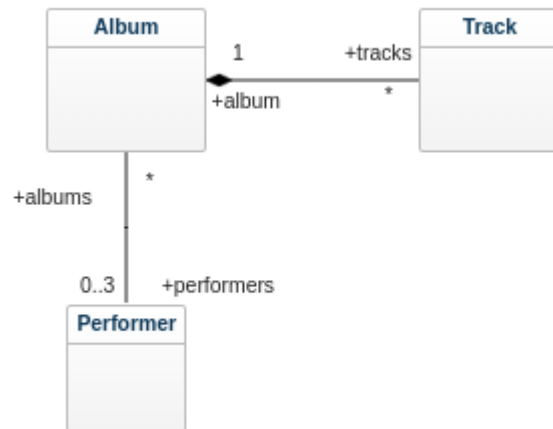


Ilustración 1. Diagrama de clases

PARTE 1. Trabajo en clase (70%)

Punto 1. Persistencia (30%)

(10%) Cree la entidad *AlbumEntity* la cual tiene un nombre (String), una carátula (String) una fecha de lanzamiento (Date), una descripción (String) y un id (UUID).

(10%) Cree la entidad *TrackEntity* la cual tiene un nombre (String), una duración (Number) y un id (UUID).

(10%) Cree la entidad *PerformerEntity* la cual tiene un nombre (String), una imagen (String), una descripción (String) y un id (UUID).



Ciclo 1 – Iteración 2

Actividad: Parcial Práctico No. 1

Punto 2. Lógica (30%)

(7.5%) Cree la clase correspondiente para la lógica de Album. Implemente los métodos **create(album)**, **findOne(id)**, **findAll()**, **delete(id)**. En el método crear valide que el nombre y la descripción del álbum no estén vacías. Un álbum no puede ser eliminado si tiene *tracks* asociados.

(7.5%) Cree la clase correspondiente para la lógica de Track. Implemente los métodos **create(albumId, track)**, **findOne()**, **findAll()**. Valide que la duración del track sea un numero positivo. No se puede crear un track si el álbum al que se va a asociar no existe.

(7.5%) Cree la clase correspondiente para la lógica de Performer. Implemente los métodos **create(performer)**, **findOne(id)**, **findAll()**. Valide que la descripción tenga como máximo 100 caracteres.

(7.5%) Cree la clase correspondiente para la asociación entre Performer y Album. Implemente el método **addPerformerToAlbum(albumId, performerId)** el cual agrega un performer a un álbum. Valide que tanto el performer como el álbum existen. Un álbum no puede tener más de tres performers asociados.

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

(10%) Implemente las pruebas para el método **create()** del servicio de Album. Asegúrese de probar por lo menos dos casos: (i) se crea el álbum correctamente, (ii) se crea un álbum con la descripción vacía. Se debe lanzar una excepción de negocio por la violación de la regla de negocio.

Entrega

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_1_clase**.

Suba el enlace del *release* como respuesta a la actividad **Parcial1** de Bloque Neón e indique allí la URL del repositorio.

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



Ciclo 1 – Iteración 2

Actividad: Parcial Práctico No. 1

PARTE 2. Trabajo en casa (30%)

Punto 1. Prueba de lógica (segunda parte) (10%)

(10%) 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 2. Controladores (10%)

(10%) Realice la implementación de los controladores NEST para exponer con una API REST los métodos generados en la capa de lógica.

Punto 3. Colecciones y pruebas de Postman (10%)

(10%) Cree las colecciones y las pruebas de Postman para el API REST creado. Exporte las colecciones a la carpeta *collections* del proyecto.

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_1_casa**.

Suba el enlace del *release* como respuesta a la actividad **Parcial1** de Bloque Neón e indique allí la URL de su repositorio.

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