

Instrucciones

1. Cree un nuevo proyecto de Nest.js
2. Abra el proyecto en el editor de su preferencia.
3. Suba el proyecto a un repositorio de GitHub en su cuenta personal.
4. Haga *commit* y *push* en su repositorio bifurcado periódicamente.

Punto 1. Persistencia (6%)

Esta aplicación tiene el propósito de crear un sistema de gestión de bibliotecas.

1. Cree la entidad *Biblioteca* en el módulo correspondiente. Una biblioteca tiene un nombre, una dirección, una ciudad y un horario de atención.
2. Cree la entidad *Libro* en el módulo correspondiente. Un libro tiene un título, un autor, una fecha de publicación y un ISBN.
3. Incluya la asociación entre *Biblioteca* y *Libro*; tenga en cuenta que una biblioteca puede tener múltiples libros y un libro puede estar disponible en distintas bibliotecas.

Punto 2. Lógica (43%)

1. Defina la lógica de *Biblioteca*, esta debe incluir los métodos *findAll*, *findOne*, *create*, *update* y *delete*. En *create* y *update*, valide que la hora de apertura sea menor a la de cierre.
2. Defina la lógica de *Libro*, esta debe incluir los métodos *findAll*, *findOne*, *create*, *update* y *delete*. En los métodos *create* y *update*, valide que la fecha de publicación sea anterior o igual a la fecha actual.
3. Defina la lógica de la asociación, esta debe incluir 5 métodos con las siguientes acciones:
 - *addBookToLibrary*: Asociar un libro a una biblioteca.
 - *findBooksFromLibrary*: Obtener todos los libros asociados a una biblioteca.
 - *findBookFromLibrary*: Obtener un libro asociado a una biblioteca.
 - *updateBooksFromLibrary*: Actualizar los libros asociados a una biblioteca.
 - *deleteBookFromLibrary*: Eliminar un libro asociado a una biblioteca.
4. Implemente las pruebas para la lógica de *Biblioteca*, para la lógica de *Libro* y para la lógica de la asociación.

MISW4403 – Diseño y Construcción de APIs
Parcial Práctico

Punto 3. API (24%)

1. Cree la clase del controlador para *Biblioteca*, agregue la ruta `/libraries` y defina los endpoints *findAll*, *findOne*, *create*, *update* y *delete* con sus respectivas anotaciones.
2. Cree la clase del controlador para *Libro*, agregue la ruta `/books` y defina los endpoints *findAll*, *findOne*, *create*, *update* y *delete* con sus respectivas anotaciones.
3. Cree la clase del controlador para la asociación *Biblioteca-Libro*, agregue la ruta de modo que se acceda a los endpoints a través de la biblioteca (ej. `/libraries/1/books/4` para *findBookFromLibrary*) e implemente los endpoints:
 - *addBookToLibrary*
 - *findBooksFromLibrary*
 - *findBookFromLibrary*
 - *updateBooksFromLibrary*
 - *deleteBookFromLibrary*

Punto 4. Pruebas de Postman (27%)

1. Defina 3 colecciones donde implemente las siguientes pruebas de Postman para las entidades y para la asociación.

Método	Bibliotecas	Libros	Bibliotecas-Libros
POST	Crear una biblioteca válida.	Crear un libro válido.	Asociar un libro a una biblioteca.
POST	Crear una biblioteca inválida.	Crear un libro inválido.	Asociar un libro que no existe a una biblioteca.
GET	Obtener todas las bibliotecas.	Obtener todos los libros.	Obtener todos los libros que están en una biblioteca.

MISW4403 – Diseño y Construcción de APIs

Parcial Práctico

GET	Obtener una biblioteca por ID.	Obtener un libro por ID	Obtener un libro asociado a una biblioteca.
GET	Obtener una biblioteca por un ID que no existe.	Obtener un libro por un ID que no existe.	Obtener un libro que no esté asociado a una biblioteca.
PUT	Actualizar una biblioteca.	Actualizar un libro.	Actualizar los libros que están asociados a una biblioteca.
PUT	Actualizar una biblioteca con un ID que no existe.	Actualizar un libro con un ID que no existe.	Actualizar los libros asociados a una biblioteca, con un libro inexistente.
DELETE	Eliminar una biblioteca por su ID.	Eliminar un libro por su ID.	Eliminar un libro previamente asociado a una biblioteca.
DELETE	Eliminar una biblioteca con un ID que no existe.	Eliminar un libro con un ID que no existe.	Eliminar un libro que no estaba previamente asociado a una biblioteca.

Entregable

- Dentro del proyecto de Nest.js cree una carpeta denominada collections y exporte ahí las colecciones.
- Suba todos los cambios a su repositorio.
- Haga un *release* con el tag v1.0.0 y el nombre parcial-practico.
- Suba el archivo .zip del *release* como respuesta a la actividad de Coursera.
- Después de finalizado el plazo de entrega no realice ninguna modificación al repositorio. Cualquier cambio, por pequeño que sea, anula automáticamente el parcial.