

## 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 consulta de cobertura de las aerolíneas.

1. Cree la entidad *Aerolinea* en el módulo correspondiente. Una aerolínea tiene un nombre, una descripción, una fecha de fundación y una página web.
2. Cree la entidad *Aeropuerto* en el módulo correspondiente. Un aeropuerto tiene un nombre, un código, un país y una ciudad.
3. Incluya la asociación entre *Aerolinea* y *Aeropuerto*; tenga en cuenta que un aeropuerto cuenta con múltiples aerolíneas y una aerolínea tiene cobertura en distintos aeropuertos.

### Punto 2. Lógica (43%)

1. Defina la lógica de *Aerolinea*, esta debe incluir los métodos *findAll*, *findOne*, *create*, *update* y *delete*. Dentro de los métodos *create* y *update*, valide que la fecha de fundación sea en el pasado, es decir que sea menor a la fecha en la que se crea.
2. Defina la lógica de *Aeropuerto*, esta debe incluir los métodos *findAll*, *findOne*, *create*, *update* y *delete*. Dentro de los métodos *create* y *update*, valide que el código del aeropuerto únicamente tenga tres caracteres.
3. Defina la lógica de la asociación, esta debe incluir 5 métodos con las siguientes acciones:
  - *addAirportToAirline*: Asociar un aeropuerto a una aerolínea.
  - *findAirportsFromAirline*: Obtener los aeropuertos que cubre una aerolínea.
  - *findAirportFromAirline*: Obtener un aeropuerto que cubre una aerolínea.
  - *updateAirportsFromAirline*: Actualizar los aeropuertos que cubre una aerolínea.
  - *deleteAirportFromAirline*: Eliminar los aeropuertos que cubre una aerolínea.

**MISW4403 - Diseño y construcción de APIs**  
**Parcial práctico**

4. Implemente las pruebas para la lógica de *Aerolinea*, para la lógica de *Aeropuerto* y para la lógica de la asociación.

### Punto 3. API (24%)

1. Cree la clase del controlador para *Aerolinea*, agregue la ruta `/airlines` y defina los endpoints `findAll`, `findOne`, `create`, `update` y `delete` con sus respectivas anotaciones.
2. Cree la clase del controlador para *Aeropuerto*, agregue la ruta `/airports` y defina los endpoints `findAll`, `findOne`, `create`, `update` y `delete` con sus respectivas anotaciones.
3. Cree la clase del controlador para la asociación *Aerolinea-Aeropuerto*, agregue la ruta de modo que se acceda a los endpoints a través de la aerolínea (ej. `/airlines/1/airports/4` para `findAirportFromAirline`) e implemente los endpoints:
  - `addAirportToAirline`
  - `findAirportsFromAirline`
  - `findAirportFromAirline`
  - `updateAirportsFromAirline`
  - `deleteAirportFromAirline`

### 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	Aerolíneas	Aeropuertos	Aerolíneas-Aeropuertos
POST	Crear una aerolínea válida.	Crear un aeropuerto válido.	Asociar un aeropuerto a una aerolínea.
POST	Crear una aerolínea inválida.	Crear un aeropuerto inválido.	Asociar un aeropuerto que no existe a una aerolínea.
GET	Obtener todas las aerolíneas.	Obtener todos los aeropuertos.	Obtener todos los aeropuertos que cubre una aerolínea.
GET	Obtener una aerolínea por ID.	Obtener un aeropuerto por ID	Obtener un aeropuerto asociado a una aerolínea.
GET	Obtener una aerolínea por un ID que no	Obtener un aeropuerto por un ID que no existe.	Obtener un aeropuerto que no esté asociado a una aerolínea.

**MISW4403 - Diseño y construcción de APIs**  
**Parcial práctico**

	existe.		
PUT	Actualizar una aerolínea.	Actualizar un aeropuerto.	Actualizar los aeropuertos que están asociados a una aerolínea.
PUT	Actualizar una aerolínea con un ID que no existe.	Actualizar un aeropuerto con un ID que no existe.	Actualizar los aeropuertos asociados a una aerolínea, con un aeropuerto inexistente.
DELETE	Eliminar una aerolínea por su ID.	Eliminar un aeropuerto por su ID.	Eliminar un aeropuerto previamente asociado a una aerolínea.
DELETE	Eliminar una aerolínea con un ID que no existe.	Eliminar un aeropuerto con un ID que no existe.	Eliminar un aeropuerto que no estaba previamente asociado a una aerolínea.

## 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 bifurcado. Cualquier cambio, por pequeño que sea, anula automáticamente el parcial.