Instrucciones

1. Cree un nuevo proyecto Usando Node.js y Nest.js
2. Abra el bifurcado 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 consultar el listado de tiendas que proveen cierto producto.

1. Cree la entidad *Producto* en el módulo correspondiente. Un producto tiene un nombre, un precio y un tipo.
2. Cree la entidad *Tienda* en el módulo correspondiente. Una tienda tiene un nombre, una ciudad y una dirección.
3. Incluya la asociación entre *Producto* y *Tienda*; tenga en cuenta que producto se puede ofrecer en múltiples tiendas y una tienda ofrece varios productos.

Punto 2. Lógica (43%)

1. Defina la lógica de *Producto*, esta debe incluir los métodos *findAll*, *findOne*, *create, update* y *delete*. Dentro de los métodos *create* y *update*, valide que el tipo del producto se encuentre dentro de los siguientes:

   Perecedero, No perecedero

2. Defina la lógica de *Tienda*, esta debe incluir los métodos *findAll*, *findOne*, *create, update* y *delete*. Dentro de los métodos *create* y *update*, valide que la ciudad de la tienda sea un código de tres caracteres (e.g., SMR, BOG, MED).
3. Defina la lógica de la asociación, esta debe incluir 5 métodos con las siguientes acciones:
   - *addStoreToProduct*: Asociar una tienda a un producto.
   - *findStoresFromProduct*: Obtener las tiendas que tienen un producto.
   - *findStoreFromProduct*: Obtener una tienda que tiene un producto.
   - *updateStoresFromProduct*: Actualizar las tiendas que tienen un producto.
   - *deleteStoreFromProduct*: Eliminar la tienda que tiene un producto.
4. Implemente las pruebas para la lógica de *Producto*, para la lógica de *Tienda* y para la lógica de la asociación.

Este documento no puede ser compartido con terceros, solo puede ser usado por aquella persona a quién fue directamente enviado.

Punto 3. API (24%)

1. Cree la clase del controlador para *Producto*, agregue la ruta `/products` y defina los endpoints *findAll*, *findOne*, *create*, *update* y *delete* con sus respectivas anotaciones.
2. Cree la clase del controlador para *Tienda*, agregue la ruta `/stores` y defina los endpoints *findAll*, *findOne*, *create*, *update* y *delete* con sus respectivas anotaciones.
3. Cree la clase del controlador para la asociación *Producto-Tienda*, agregue la ruta de modo que se acceda a los endpoints a través del producto (ej. `/products/1/stores/4` para *addStoreToProduct*) e implemente los endpoints:
    - *addStoreToProduct*
    - *findStoresFromProduct*
    - *findStoreFromProduct*
    - *updateStoresFromProduct*
    - *deleteStoresFromProduct*

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	Producto	Tienda	Asociación
POST	Crear un producto válido.	Crear una tienda válida.	Agregar una nueva tienda a las oferentes de un producto.
POST (error)	Crear un producto inválido.	Crear una tienda inválida.	Asociar una tienda que no existe a las oferentes de un producto.
GET	Obtener todos los productos.	Obtener todas las tiendas.	Obtener todas las tiendas que ofrecen un producto.
GET	Obtener un producto por ID.	Obtener una tienda por ID	Obtener una tienda que ofrece un producto.
GET (error)	Obtener un producto por un ID que no existe.	Obtener una tienda por un ID que no existe.	Obtener una tienda que ofrece un producto que no existe.

Este documento no puede ser compartido con terceros, solo puede ser usado por aquella persona a quién fue directamente enviado.

PUT	Actualizar un producto.	Actualizar una tienda.	Actualizar las tiendas que ofrecen un producto.
PUT (error)	Actualizar un producto con un ID que no existe.	Actualizar una tienda con un ID que no existe.	Actualizar las tiendas que ofrecen un producto, con una tienda inexistente.
DELETE	Eliminar un producto por su ID.	Eliminar una tienda por su ID.	Eliminar una tienda que ofrece un producto.
DELETE (error)	Eliminar un producto con un ID que no existe.	Eliminar una tienda con un ID que no existe.	Eliminar una tienda que no ofrece un producto.

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 prueba-practica.
- Envie el archivo .zip del *release* como respuesta al correo enviado con este documento.
- Después de realizado el release no realice ninguna modificación al repositorio bifurcado. Cualquier cambio, por pequeño que sea, anula automáticamente la prueba.

Este documento no puede ser compartido con terceros, solo puede ser usado por aquella persona a quién fue directamente enviado.