Proyecto final - Back End

Se desarrollará un servidor que contenga los endpoints y servicios necesarios para gestionar los productos y carritos de compra en el e-commerce

Primer entrega.

- FILMINA 1:
  - [x] ~~Desarrollar el servidor basado en Node.JS y express, que escuche en el puerto 8080.~~
  - [x] ~~Tiene que disponer de dos grupos de rutas:~~
    - RUTAS:
      - [x] ~~/products .js~~
      - [x] ~~/carts.js~~
  - [x] ~~Dichos endpoints estarán implementados con el router de express, con las siguientes especificaciones:~~

PRODUCTS.JS:

- [x] ~~Para el manejo de productos, tendrá su router en /api/products~~
- [x] ~~Configurar las siguientes rutas:~~
  - RUTAS:

- [x] ~~La ruta raíz GET / deberá listar todos los productos de la base.~~
  - [x] ~~debe incluir la limitación ?limit del desafío anterior.(ESTO NO SE SI ESTA BIEN)~~
- [x] ~~La ruta GET /:pid deberá traer sólo el producto con el id proporcionado.~~

▼ FILMINA 2:

PRODUCTS.JS:

▼ RUTAS:

- [x] ~~La ruta raíz POST / deberá agregar un nuevo producto con los campos:~~

▼ CAMPOS:

- Todos los campos son obligatorios, a excepción de thumbnails.
- [x] ~~id: Number/String~~
  - (A tu elección, el id NO se manda desde body, se autogenera como lo hemos visto desde los primeros entregables, asegurando que NUNCA se repetirán los ids en el archivo.)
- [x] ~~title:String~~
- [x] ~~description:String~~
- [x] ~~code:String~~
- [x] ~~price:Number~~
- [x] ~~status:Boolean~~
  - Status es true por defecto.
- [x] ~~stock:Number~~
- [x] ~~category:String~~
- [x] ~~thumbnails~~
  - Array de Strings que contenga las rutas donde están almacenadas las imágenes referentes a dicho producto

▼ FILMINA 3:

PRODUCTS.JS:

- [x] ~~La ruta PUT /:pid deberá tomar un producto y actualizarlo por los campos enviados desde body.~~
    - NUNCA se debe actualizar o eliminar el id al momento de hacer dicha actualización.(esto no lo entendi)
- [x] ~~La ruta DELETE /:pid deberá eliminar el producto con el pid indicado.~~

CARTS.JS:

- [x] ~~El carrito, tendrá su router en /api/carts/~~
- [ ] Configurar dos rutas: (TE QUEDASTE ACA)

▼ RUTAS:

- [x] ~~**La ruta raíz POST / deberá crear un nuevo carrito con la siguiente estructura:**~~

▼ ESTRUCTURAS DEL POST/:

- [x] ~~Id:Number/String~~
    - (A tu elección, de igual manera como con los productos, debes asegurar que nunca se dupliquen los ids y que este se autogenere).
- [x] ~~products: Array~~
    - contendrá objetos que representen cada producto

▼ FILMINA 4:

CARTS.JS:

▼ RUTAS:

- [x] ~~**La ruta GET /:cid deberá listar los productos que pertenezcan al carrito con el parámetro cid proporcionados.**~~
- [x] ~~**La ruta POST /:cid/product/:pid deberá agregar el producto al arreglo "products" del carrito seleccionado, agregándose como un objeto bajo el siguiente formato:**~~

▼ FORMATO:

- ~~Product: SÓLO DEBE CONTENER EL ID DEL PRODUCTO (Es crucial que no agregues el producto completo)~~
- ~~Quantity: debe contener el número de ejemplares de dicho producto. El producto, de momento, se agregará de uno en uno.~~
- ~~Además, si un producto ya existente intenta agregarse al producto, incrementar el campo quantity de dicho producto.~~

▼ FILMINA 5:

PERSISTENCIA:

- ~~La persistencia de la información se implementará utilizando el file system.~~
- ~~los archivos “productos,json” y “carrito.json”, respaldan la información.~~
- No es necesario realizar ninguna implementación visual, todo el flujo se puede realizar por Postman o por el cliente de tu preferencia.