**Consigna**

Close

**Entrega Final**

**Objetivos generales**

* Contarás con Mongo como sistema de persistencia principal
* Tendrás definidos todos los endpoints para poder trabajar con productos y carritos.

**Objetivos específicos**

* Profesionalizar las consultas de productos con filtros, paginación y ordenamientos
* Profesionalizar la gestión de carrito para implementar los últimos conceptos vistos.

**Formato**

* Link al repositorio de Github, sin la carpeta de node\_modules

**Sugerencias**

* Permitir comentarios en el archivo
* La lógica del negocio que ya tienes hecha no debería cambiar, sólo su persistencia.
* Los nuevos endpoints deben seguir la misma estructura y lógica que hemos seguido.

Video explicativo entrega final: <https://drive.google.com/file/d/1nQUXoZ7Oq0uGukaE13PL-E6dM77KjwNv/view?usp=sharing>

**Se debe entregar**

* Con base en nuestra implementación actual de productos, modificar el método GET / para que cumpla con los siguientes puntos:
  + Deberá poder recibir por query params un limit (opcional), una page (opcional), un sort (opcional) y un query (opcional)
    - limit permitirá devolver sólo el número de elementos solicitados al momento de la petición, en caso de no recibir limit, éste será de 10.
    - page permitirá devolver la página que queremos buscar, en caso de no recibir page, ésta será de 1
    - query, el tipo de elemento que quiero buscar (es decir, qué filtro aplicar), en caso de no recibir query, realizar la búsqueda general
    - sort: asc/desc, para realizar ordenamiento ascendente o descendente por precio, en caso de no recibir sort, no realizar ningún ordenamiento
    - query, el tipo de elemento que quiero buscar (es decir, qué filtro aplicar), en caso de no recibir query, realizar la búsqueda general
    - sort: asc/desc, para realizar ordenamiento ascendente o descendente por precio, en caso de no recibir sort, no realizar ningún ordenamiento

**Se debe entregar**

* El método GET deberá devolver un objeto con el siguiente formato:

{

status:success/error

payload: Resultado de los productos solicitados

totalPages: Total de páginas

prevPage: Página anterior

nextPage: Página siguiente

page: Página actual

hasPrevPage: Indicador para saber si la página previa existe

hasNextPage: Indicador para saber si la página siguiente existe.

prevLink: Link directo a la página previa (null si hasPrevPage=false)

nextLink: Link directo a la página siguiente (null si hasNextPage=false)

}

* Se deberá poder buscar productos por categoría o por disponibilidad, y se deberá poder realizar un ordenamiento de estos productos de manera ascendente o descendente por precio.

**Se debe entregar**

* Además, agregar al router de carts los siguientes endpoints:
  + DELETE api/carts/:cid/products/:pid deberá eliminar del carrito el producto seleccionado.
  + PUT api/carts/:cid deberá actualizar todos los productos del carrito con un arreglo de productos.
  + PUT api/carts/:cid/products/:pid deberá poder actualizar SÓLO la cantidad de ejemplares del producto por cualquier cantidad pasada desde req.body
* DELETE api/carts/:cid deberá eliminar todos los productos del carrito
* Esta vez, para el modelo de Carts, en su propiedad products, el id de cada producto generado dentro del array tiene que hacer referencia al modelo de Products. Modificar la ruta /:cid para que al traer todos los productos, los traiga completos mediante un “populate”. De esta manera almacenamos sólo el Id, pero al solicitarlo podemos desglosar los productos asociados.

**Se debe entregar**

* Modificar la vista index.handlebars en el router de views ‘/products’, creada en la pre-entrega anterior, para visualizar todos los productos con su respectiva paginación. Además, cada producto mostrado puede resolverse de dos formas:
  + Llevar a una nueva vista con el producto seleccionado con su descripción completa, detalles de precio, categoría, etc. Además de un botón para agregar al carrito.

Sugerencia de la ruta: “/products/:pid”.

* Contar con el botón de “agregar al carrito” directamente, sin necesidad de abrir una página adicional con los detalles del producto.
* Además, agregar una vista en ‘/carts/:cid (cartId) para visualizar un carrito específico, donde se deberán listar SOLO los productos que pertenezcan a dicho carrito.
* [Criterios de evalución](https://drive.google.com/file/d/1eUCxiLqQY4mVp7cxZ2khAs3ZY2ii--MW/view?usp=drive_link)