

Desarrollo Web Full Stack Node

Ejercitación - M03C12

Mercado Liebre v3 - CRUD

Introducción

Vamos a incorporar a nuestro ecommerce la posibilidad leer productos desde una base de datos (por ahora json), agregarlos, borrarlos y modificarlos.

El conjunto de estas funcionalidades es conocido como ABM (Alta, Baja, Modificación) o por sus siglas en inglés: CRUD (Create, Read, Update Delete).

Requisitos

- **HTML / CSS resuelto:** en clases pasadas desarrollamos la Home y la sección de Detalle del Producto en HTML y CSS. Por lo tanto deberías contar con ellas ahora. Sin embargo, si nuevamente *"tuviste problema con los archivos"* o *"el perro volvió a comerse tu tarea"*, dejamos [estos archivos](#) base en donde encontrarás todo el HTML y el CSS, con eso tu enfoque estará 100% puesto en el desarrollo de CRUD..

Objetivo

Para esta versión necesitamos contar con la posibilidad de acceso a las urls detalladas. Para darte una mano, en cada caso aclaramos cual es el método http que debemos usar. Para simplificarlo en esta etapa vamos a usar únicamente “GET” y “POST”.

- / (GET)
 - Se deberán mostrar los productos separados en dos secciones. Los últimos visitados y los productos en oferta
- /products (GET)
 - Se deberán listar todos los productos presentes en el base de datos json.
- /products/detail/:id (GET)
 - Detalle de producto. Se deberá mostrar el producto con el correspondiente. Cada producto deberá contar con dos (2) botones de acción: BORRAR y MODIFICAR.
 - Botón BORRAR: eliminará al producto de la DB Json. (*POST*)
 - Botón MODIFICAR: modificará al producto correspondiente en la DB Json. (TIP: */products/edit/:id - Viajará por GET Y /products/edit - Viajará por POST*)
- /products/create (GET)
 - Mostrará el formulario de creación para un producto.. (TIP: */products/create - Viajará por POST*)

¡Atención! Algunas de las rutas anteriores pueden no ser accesibles desde la barra de direcciones del navegador. Deberás fijarte qué método se usa en cada endpoint.

Consignas

A continuación se detalla lo que deberá mostrar cada ruta.

1. Sobre la interfaz

Como lo mencionamos unos párrafos antes. La interfaz ya está resuelta. Tu trabajo será aportar las nuevas funcionalidades.

2. Sobre el desarrollo

La estructura de la aplicación cambió un poco, así que te recomendamos que la revises antes de comenzar a codear. Para ello, preguntanos por qué de esta nueva organización.

3. Sobre los requerimientos técnicos

La aplicación **deberá correr sobre un entorno hecho con Node y Express**.

El desarrollo **seguirá estando basado en EJS** como motor de plantillas y bajo un sistema de archivos parciales.

Se espera que **si un producto tiene descuento** el precio final que se muestra sea: $\text{precioReal} - \text{descuentoDelProducto}$. Y que adicionalmente se muestre el porcentaje de descuento aplicado.

Cada producto en la Home **deberá llevar a la sección de detalle** del producto correspondiente. En la sección Detalle de producto se deberá mostrar:

- Pestaña del navegador con el nombre de producto.
- Nombre del producto.
- Imagen del producto.

- Descripción del producto.
- Precio real, descuento y precio final.
- Si el producto no tiene descuento, mostrar sólo el precio real.
- Los precios deberán mostrarse con separador de miles.

En la a sección de Detalle de producto se deberá incluir los dos botones de acción para cada producto. MODIFICAr y BORRAR.

Al borrar un producto se quiere que el mismo sea eliminado de la base de datos Json.

No deberán existir productos con el mismo id.

Teniendo en claro todo esto, te recomendamos que revises en detalle todos los archivos antes de arrancar para saber exactamente *qué parte podrás tocar y qué partes no debes tocar*.

¡Mucha suerte!

Desafío extra

Felicitaciones por haber completado toda la práctica. Ahora tenemos un desafío extra. Como vimos en el material virtual tenemos otros métodos semánticamente más claros parar acciones por "POST". Te invitamos a que implementes (o revises si ya está implementado) dentro del ejercicio lo necesario para usar los métodos "PUT" y "DELETE" y luego los apliques en las rutas que correspondan. Si todo va bien obtendrás el mismo funcionamiento ahora con los 4 métodos: GET, POST, PUT, DELETE.