

## Desarrollo Web Full Stack Node

Ejercitación - M03C12

# Mercado Liebre v3 - CRUD

### Introducción

Todo **eMarket** que se respete necesita que sea posible Agregar, Borrar y Modificar productos y que los mismos se persistan en una base de datos. Adicionalmente también debe permitir Leer los productos presentes en dicha base de datos de una manera rápida y sencilla.

A este sistema en el mundo del desarrollo se le conoce como ABM (Alta, Baja, Modificación) o CRUD (Create, Read, Update Delete).

Así que tu enfoque en esta oportunidad será desarrollar dicha funcionalidad, pues nuestros inversores quieren salir con la web al aire lo antes posible.

## Requisitos

- **HTML / CSS resuelto:** en la versión anterior del proyecto, se te solicitó desarrollar la Home y la sección de Detalle del Producto en HTML y CSS. Por lo tanto deberías tener resuelta la misma. Sin embargo, somos conscientes que por ahí *"tuviste problema con los archivos"* o que *"el perro se te comió la tarea"*, por ello te dejamos [estos archivos](#) base en donde encontrarás TODO lo referente al HTML y CSS, con eso tu enfoque estará 100% puesto en el desarrollo de CRUD..

## Objetivo

Para la construcción de esta versión del sitio web, el cliente espera contar con la posibilidad de acceso a las siguientes URL's:

- / (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. (*TIP: [/products/delete/id](#) - Viajará por DELETE*)
  - Botón MODIFICAR: modificará al producto correspondiente en la DB Json. (*TIP: [/products/edit/:id](#) - Viajará por GET Y [/products/edit](#) - Viajará por PUT*)
- /products/create (GET)
  - Mostrará el formulario de creación para un producto.. (*TIP: [/products/create](#) - Viajará por POST*)

**¡Atención!** Algunos de los anterior endpoint no será accesibles desde la barra de direcciones del navegador, deberá 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 el dinamismo.

### 2. Sobre el desarrollo

La estructura de la aplicación cambió un poco, así que te recomendamos que la revisés antes de comenzar a codear. Para ello, pregúntale a tu docente el por qué de esta nueva organización.

### 3. Sobre los requerimientos técnicos

Nuestro cliente y su consultor de tecnología nos recomendaron altamente que:

- El desarrollo **seguirá estando basado en EJS** como motor de plantillas y bajo un sistema de archivos parciales.
- Nuevamente te hacen énfasis respecto al precio de los productos, pues el mismo **deberá usar un separador de miles**,. Es por ello que se te provee de esta función para tal fin:

```
const toThousand = n =>
n.toString().replace(/\B(?=(\d{3})+(?!\d))/g,
".");
```

- 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.
- La aplicación **deberá correr sobre un entorno hecho con Node y Express**. Pues esto les provee escalabilidad, ya que si todo marcha como es debido, piensan sumar más secciones y características al proyecto en un futuro cercano.

Teniendo en claro todo esto, te recomendamos que revisés todos los archivos que te fueron suministrados y que los revisés en detalle antes de arrancar, así de esa manera saber exactamente *qué si podés tocar y qué no debés tocar*.

¡Mucha suerte!