

Práctica de Front-end

Crear una aplicación Front-end

En esta práctica añadiremos el componente que le faltaba a las prácticas anteriores la interactividad que nos provee JavaScript para manipular el DOM. De esta forma, vamos a crear una aplicación funcional del lado del cliente (front-end). Además, ahora podremos realizar solicitudes de HTTP para realizar operaciones de altas, bajas, cambios y consultas.

Descripción de la práctica

Completaremos la parte del front-end de la aplicación de e-commerce. Esto implicará realizar lo siguiente.

1. Hacer una petición al back-end de los productos y mostrar el listado de productos.
2. Agregar los productos actuales al carrito de compras.
3. Leer la lista de productos en el carrito de compras (solicitar a back-end).
4. Cargar el carrito de compras con los datos del servidor / sessionStorage.
5. Modificar los productos del carrito de compras.
6. Eliminar productos del carrito de compras.
7. CRUD de productos.

A todo esto, se le añade que se deben activar/desactivar botones o inputs en algunos momentos, ocultar o mostrar elementos del HTML, solicitar confirmación para realizar una acción, etc.

Antes de iniciar

Usa tu código de la práctica 1. Podrás quitar los botones que no se necesitan y añadir nuevos en caso de necesitarse. Tienes libertad de modificar el html siempre que cumplas con los requerimientos. **Todo el código que realices debe estar en archivos externos de JavaScript. No olvides importarlos en el html. Se recomienda que cada html tenga su archivo .js para facilitar las cosas.**

Recuerda que para poder cargar las imágenes, hojas de estilos y los controladores de Frontend de Javascript debes de poder hacer cargas de archivos estáticos. Para esto, debes modificar tu servidor de Node, en el archivo server.js:

```
app.use(express.static('app'));  
app.use('/views', express.static('views'));
```

Considera que debes de ajustar y cambiar las rutas de a cuerdo a la jerarquía de archivos que tú tengas. Puedes ver cómo funciona la librería de archivos estáticos aquí:

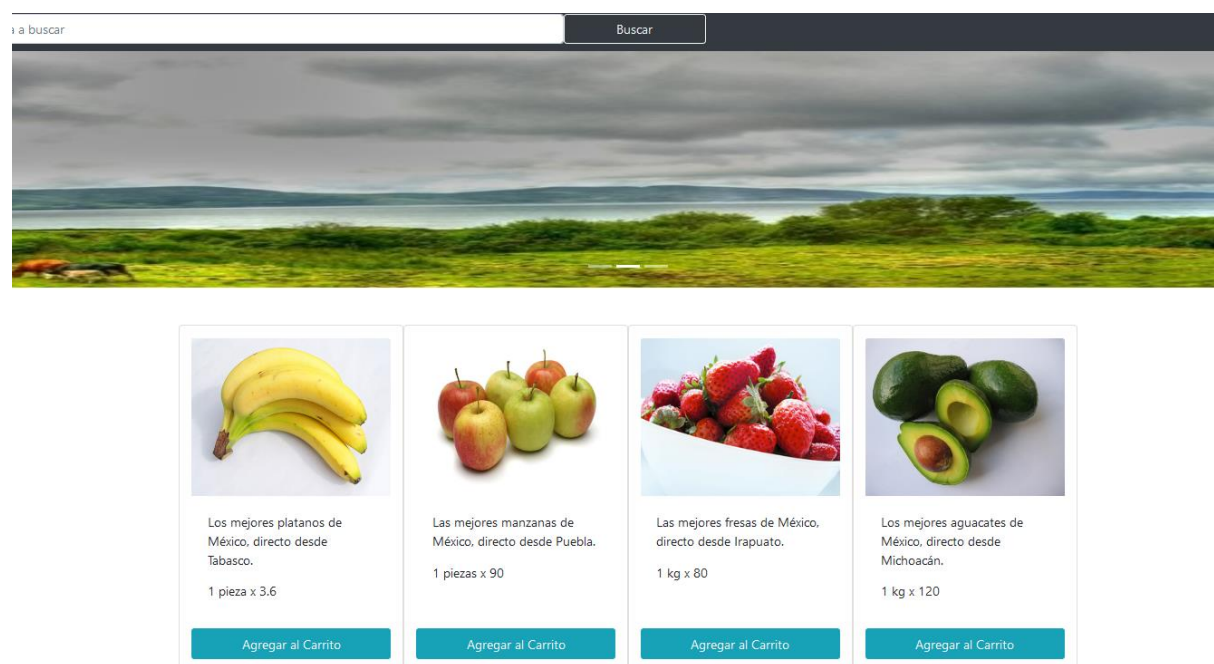
<https://www.geeksforgeeks.org/how-to-fetch-images-from-node-js-server/>

Mostrar la lista de productos

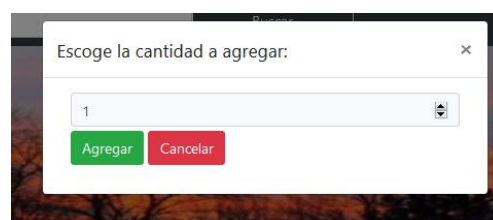
Lo primero que debemos hacer es conectar nuestra página principal (Home), la cual contiene el listado de productos. Para esto nos apoyaremos de nuestro **Endpoint** en el servidor para leer la lista de productos. Recuerda que para hacer llamadas de **HTTP** desde el front-end debemos usar las librerías de **AJAX**.

Posteriormente tendremos que mostrar nuestra lista de productos usando **DOM**, si realizaste la parte opcional de la práctica anterior apóyate de las funciones para convertir a **HTML**, de lo contrario añádelas para poder mostrar la lista de productos.

Es importante tratar de usar el paginado para cargar unicamente de 4 en 4 los productos, apoyate de **DOM** para verificar en que pagina esta el usuario y carga el subarreglo correspondiente.



Asegúrate de añadir un botón para agregar al carrito a cada elemento y que el mismo te abra un modal para seleccionar la cantidad a agregar.



Agregar Productos al Carrito de Compras

Para hacer el manejo del carrito de compras nos apoyaremos de la memoria del navegador para guardar la información correspondiente. Investiga cómo usar **sessionStorage** y agrega en un archivo de .js algunas funciones para manejar el carrito de compras. Se recomienda crear uno para leer, uno para borrar y otro para inicializar.

Una vez que tengas estos métodos agrega los eventos necesarios a tus tarjetas de productos para que, al darle agregar en el modal, se actualice correctamente el carrito de compras que está guardado en **sessionStorage**.

Redireccionando al Carrito de Compras













Al dar clic en el carrito de compras, deberás cargar la lista de productos correspondientes. Para esto, apóyate del endpoint correspondiente y asegúrate de actualizar la información en el sessionStorage. Si la respuesta es exitosa debes redirigir a la sección de carrito de compras usando **window.location.href = "url"**.

Cargar el Carrito de Compras

Al cargar esta página debes mostrar todos los productos guardados en el carrito de compras, para esto puedes hacer algo similar a lo que se realizó para la página principal.

Asegúrate de cargar el total de compras correctamente, así como toda la información correspondiente a cada producto.

Resumen de Compra

<div><div>Manzana </div><div><div>Cantidad:</div><div>5</div><div></div></div><div><div>Precio:</div><div>90.00</div><div>\$ m.n.</div></div></div> <div></div>
<div><div>Aguacate </div><div><div>Cantidad:</div><div>6</div><div></div></div><div><div>Precio:</div><div>120.00</div><div>\$ m.n.</div></div></div> <div></div>
<div><div>Platano </div><div><div>Cantidad:</div><div>1</div><div></div></div><div><div>Precio:</div><div>3.60</div><div>\$ m.n.</div></div></div> <div></div>

Total de Compra

Manzana: 5 x 90.00

Aguacate: 6 x 120.00

Platano: 1 x 3.60

Monto a pagar: 1173.60


Pagar

Cancelar

Editar Producto




Al presionar el botón de editar (icono de lápiz), debemos habilitar el input de cantidad y mostrar 2 botones nuevos (quitar el lápiz), uno para cancelar y otro para confirmar el cambio.

Asegúrate de mantener el valor original si se cancela la edición, además deberás actualizar el carrito de compras con el nuevo valor en caso de confirmar el cambio. Recuerda validar en caso de que la cantidad sea 0 (eliminar) o algún valor negativo.

Manzana 

Cantidad:


5



Precio:

90.00

\$ m.n.



Eliminar Producto del Carrito de Compras

Añade un manejador de eventos al botón de borrado y asegúrate de eliminar el elemento correspondiente del carrito de compras. Recuerda actualizar el **HTML** para que se muestre el cambio.

Administración de Productos

Esta parte es opcional, puedes crear una nueva página de [HTML](#), donde le des acceso al administrador para agregar nuevos productos. También debes permitir ver los productos actuales y modificar / eliminarlos según corresponda.

El formato es libre, pero se deben conectar correctamente todas las operaciones del CRUD de productos.

Evaluación

En la entrega debes mandar todos tus archivos y una tabla donde indicas cuales requerimientos has cumplido. Además, debes indicar la ruta para revisar tu aplicación en internet.

Requerimiento	Valor	Realizado (Si/No)
1. Se muestra el listado de productos correctamente con los elementos del servidor.	15	
2. El paginado permite cargar 4 productos por página.	10	
3. Es posible agregar elementos al carrito de compras.	15	
4. El carrito de compras contiene la información necesaria para hacer los cálculos del total de compra.	10	
5. Al entrar a la ventana de carrito de compras se cargan correctamente los productos agregados y se muestra el total de compra actualizado.	15	
6. Al dar clic en el botón de editar se habilita el input y se muestra la opción de confirmar o cancelar el cambio.	15	

7. Al dar clic en el botón de borrar se elimina el elemento del carrito de compras	10	
8. El carrito de compras se guarda correctamente en el almacenamiento del navegador (sessionStorage)	10	
9. Existe una sección para agregar productos nuevos (se debe validar que se llenen los campos correspondientes), además se permite la edición y borrado de productos existentes.	25	

Si sacas más de 100 se suma la calificación a prácticas anteriores.

En el documento que entregues tu tabla con lo que hiciste y lo que no, también pon una reflexión sobre la práctica.