



Diseño y Programación de Software Multiplataforma

Instructor: Ing. Alexander Alberto Sigüenza Campos

Integrantes: Fredy Antonio Alfaro Santos
AS231934

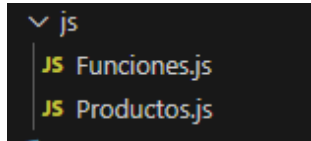
Fecha de entrega: 03 de marzo de 2024

Live: <https://fredy34009.github.io/Carrito/index.html>

Link Del video: https://udbedu-my.sharepoint.com/:f:/g/personal/as231934_alumno_udb_edu_sv/Ek6U_k_1g8pGkUnwCi-tOdcBlf65B5gh2pl3Phe6_Fugww?e=CO1esB

Organización del Código

Módulos: Son estructuras de archivos que contienen funciones específicas, en la App se usan dos módulos alojados en la carpeta js



Funciones: Son un conjunto de instrucciones que permiten realizar una tarea en específico ya sea realizar cálculos, retornar una salida etc.

```
function calcularTotal() {
  return carrito.reduce((total, item) => {
    const miItem = listaProductos.filter((itemBaseDatos) => {
      return itemBaseDatos.id === parseInt(item);
    });
    return total + miItem[0].precio;
  }, 0).toFixed(2);
}
```

Manipulación del DOM

Manipulación del DOM: permite a los desarrolladores usar la lógica de programación para realizar cambios en el código HTML. Es una manera confiable de realizar cambios que convierten los sitios web estáticos en dinámicos.

```
const DOMitems = document.querySelector('#items');
const DOMcarrito = document.querySelector('#carrito');
const DOMtotal = document.querySelector('#total');
const DOMbotonVaciar = document.querySelector('#boton-vaciar');
const DOMbotonComprar = document.querySelector('#boton-comprar');
const DOMtextCarrito = document.querySelector("#textCarrito");
const DOMtextTotal = document.querySelector("#textTotal");
/**

var modal = document.getElementById("ventanaModal");
var modalCancelar = document.getElementById("boton-cancelar");
var modalAceptar = document.getElementById("boton-aceptar");
```

Actualización del DOM: Es todo lo que sucede después de hacer referencia y seleccionar los elementos con los que se desea trabajar. Esto conduce a un cambio en el estado del elemento, de estático a dinámico. En el siguiente ejemplo se muestra una función que actualiza el DOM mostrando y ocultando dos botones el total y un texto

```
function mostrarBotonCompras() {  
  if (carrito.length <= 0) {  
    DOMtextTotal.style.display = 'none';  
    DOMbotonComprar.style.display = 'none';  
    DOMbotonVaciar.style.display = 'none';  
    DOMtextCarrito.innerHTML = 'Añadir productos';  
  }  
  else {  
    DOMbotonComprar.style.display = 'block';  
    DOMbotonVaciar.style.display = 'block';  
    DOMtextCarrito.innerHTML = 'Carrito';  
    DOMtextTotal.style.display = 'block';  
  }  
}
```

Manejo de Eventos

Eventos: Los eventos se lanzan dentro de la ventana del navegador y usualmente están asociados a un elemento en específico dentro de dicha ventana. Esto puede ser un solo elemento, un grupo de elementos

```
modalAceptar.addEventListener("click", function () {  
  modal.style.display = "none";  
  vaciarCarrito();  
  Swal.fire({  
    title: 'Exito',  
    text: 'Compra exitosa Gracias',  
    icon: 'success'  
  })  
});
```

Evento 'click' al dar clic al botón Aceptar del modal

Uso de Arrays y Objetos

Representar cada producto como un objeto

```
const listaProductos = [
  {
    id: 1,
    nombre: 'Audifonos Sony',
    precio: 15.50,
    imagen: './image/audifonos.jpeg'
  },
  {
    id: 2,
    nombre: 'Samsung A24',
    precio: 125.99,
    imagen: './image/samsungA24.png'
  },
  {
    id: 3,
    nombre: 'Hylou RT',
    precio: 60.00,
    imagen: './image/smartwachtRT.jpg'
  },
  {
    id: 4,
    nombre: 'Iphone 15',
    precio: 1599.00,
    imagen: './image/iphone 15.jpg'
  },
  {
    id: 5,
    nombre: 'Tab A9',
    precio: 319.00,
    imagen: './image/tabA9.png'
  },
  {
    id: 6,
    nombre: 'MacBook Air',
    precio: 1269.00,
    imagen: './image/macpro.jpg'
  }
];
```

Cálculos y Actualizaciones Dinámicas

Calcular el total del carrito y los subtotales por producto en tiempo real.

Carrito

1 x Hylou RT -\$60 \$60.00

X

Total: \$ 60.00

Vaciar

Pagar

Carrito

2 x Hylou RT -\$60 \$120.00

X

Total: \$ 120.00

Vaciar

Pagar

Actualizar la interfaz de usuario al agregar, eliminar o modificar productos en el carrito.

Carrito

1 x MacBook Air -\$1269 \$1269.00

X

Total: \$ 1269.00

Vaciar

Pagar

Carrito

2 x Hylou RT -\$60 \$120.00

X

1 x MacBook Air -\$1269 \$1269.00

X

Total: \$ 1389.00

Vaciar

Pagar

Facturación

×

Detalles de la factura

Unidades	Descripcion	P/U	Total
2	Hylou RT	\$60	\$120
1	Samsung A24	\$125.99	\$125.99
1	Tab A9	\$319	\$319
1	Iphone 15	\$1599	\$1599
1	MacBook Air	\$1269	\$1269

Subtotal: 3432.99


Cancelar

Pagar


Iva(13%): 446.29

Total: \$3879.28

Estilo y Diseño




Tienda Electronica SmartShop



Audifonos Sony

\$ 15.5


Añadir al carrito



Samsung A24

\$ 125.99


Añadir al carrito



Hylou RT

\$ 60


Añadir al carrito



Iphone 15

\$ 1599


Añadir al carrito



Tab A9

\$ 319

Añadir al carrito



MacBook Air

\$ 1269

Añadir al carrito

Carrito

1 x Hylou RT -\$60 \$60.00

X

1 x MacBook Air -\$1269 \$1269.00

X

1 x Tab A9 -\$319 \$319.00

X

Total: \$ 1648.00

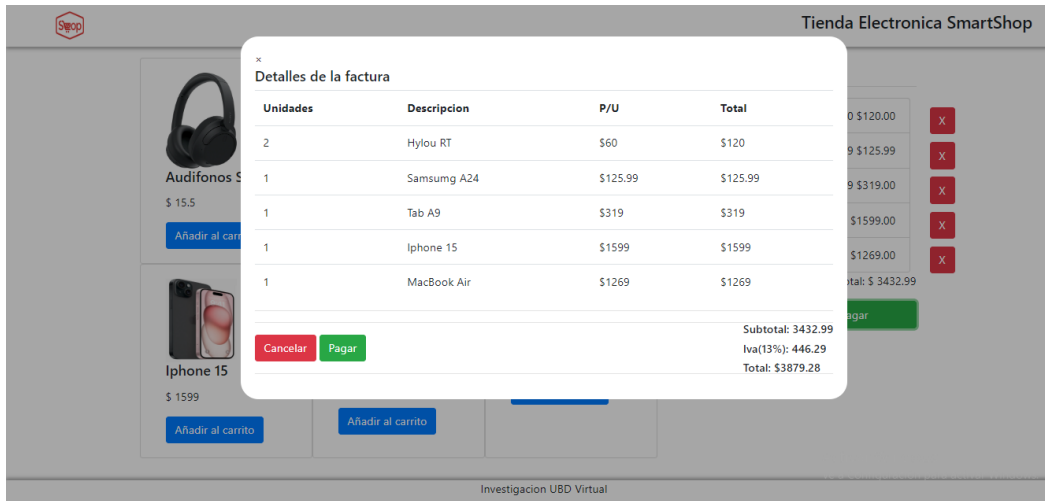
Vaciar

Pagar

Activar Windows

Ve a Configuración para activar Windows

Investigacion UBD Virtual



Documentación

Comentarios

```
//Funcion para borrar un elemento del carrito
function borrarItemCarrito(evento) {
  const id = evento.target.dataset.item;
  carrito = carrito.filter((carritoId) => {
    return carritoId !== id;
  });
  renderizarCarrito();
  mostrarBotonCompras();
}

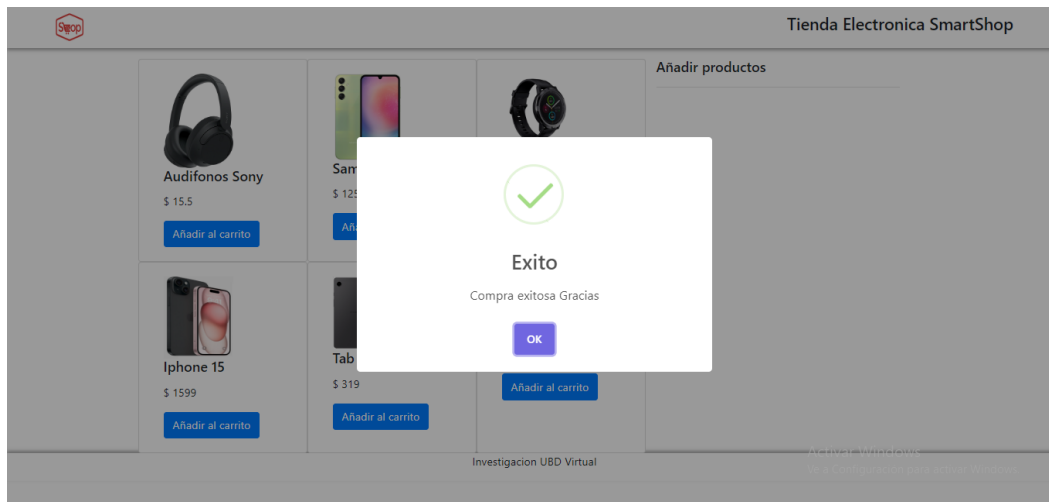
/**
 * Calcula el precio total teniendo en cuenta los productos repetidos
 */
function calcularTotal() {
  return carrito.reduce((total, item) => {
    const miItem = listaProductos.filter((itemBaseDatos) => {
      return itemBaseDatos.id === parseInt(item);
    });
    return total + miItem[0].precio;
  }, 0).toFixed(2);
}

//Vacía el carrito y vuelve a dibujarlo
function vaciarCarrito() {
  carrito = [];
  renderizarCarrito();
  mostrarBotonCompras();
}

// Funcion para realizar la compra muestra la factura
function comprar() {
```

Creatividad

Uso de SweetAlert2: se usó para crear diálogos personalizados



Se ha diseñado un carrito de compras en JavaScript en el cual es posible agregar productos, eliminar, vaciar el carrito de compras, visualizar una factura y ver el total de la compra, se ha utilizado la implementación de SweetAlert2 para crear diálogos personalizados para una experiencia de usuario agradable.