UNIVERSIDAD DON BOSCO VIRTUAL

Diseño y Programación de Software Multiplataforma

DPS941 G01T (Virtual)

TAREA

Desafío 1: "Proyecto de carrito de compras en JavaScript con Facturación"

INTEGRANTES

Jairo José Hernández Abrego HA190640

Hernández Rivas, Luis Gustavo HR233189

Helen Tatiana Barrera Guzmán BG240719

Darwin Ezequiel Martínez Rosales MR251336

Vladimir Alexander Ayala Cabrera AC202262

sábado, 28 de febrero de 2026

Explicación de la estructura del código:

El proyecto está hecho en base a un kiosko de gasolinera ficticia llamada “Del Bienestar”. El desafío está estructurado de manera modular utilizando todas las técnicas de programación solicitadas, subido a un repositorio de GitHub:

La carpeta llamada “css” posee el archivo “style.css” para garantizar que el entorno visual del ejercicio esté correcto y sea agradable para el usuario.

Pequeña demostración del código de style.ccs

Luego tenemos la carpeta “js” en donde se encuentra el archivo “app.js” aquí se tienen todas las clases para agregar productos al carrito y su debido stock, así como una función para cargar dicho inventario y luego poder ser usado en el index.html a manera visual para que el usuario pueda escoger los productos que desee:

Kiosko-DPS / js / app.js

Jairo090101 Agregando cambios en diseño de facturacion e historico

Code Blame 235 lines (197 loc) · 8.57 KB

```
// app.js
//Elaboración de la clase Producto y el inventario del kiosco
class Producto {
    constructor(id, icono, nombre, stock, precio) {
        this.id = id;
        this.icono = icono;
        this.nombre = nombre;
        this.stock = stock;
        this.precio = precio;
    }
}

// cargar inventario de localStorage si existe, de lo contrario usar valores iniciales
let inventarioKiosco = [];
function cargarInventario() {
    const datos = localStorage.getItem('inventario');
    if (datos) {
        // reconstruir objetos Producto
        const parsed = JSON.parse(datos);
        inventarioKiosco = parsed.map(p => new Producto(p.id, p.icono, p.nombre, p.stock, p.precio));
    } else {
        inventarioKiosco = [
            new Producto(1, "☕", "Café Americano", 15, 1.50),
            new Producto(2, "⚡", "Bebida Energética", 24, 2.75),
            new Producto(3, "🧀", "Nachos con Queso", 18, 2.50),
            new Producto(4, "🍩", "Donas Glaseadas", 12, 1.00),
            new Producto(5, "🍟", "Churritos Picantes", 20, 0.75),
            new Producto(6, "💧", "Agua Mineral", 30, 1.25),
            new Producto(7, "🥜", "Maní Salado", 25, 0.50),
            new Producto(8, "🍪", "Galletas Surtidas", 20, 0.85),
            new Producto(9, "🍭", "Paletas de Dulce", 15, 0.60),
            new Producto(10, "🚬", "Caja de Cigarros", 10, 4.50),
            new Producto(11, "🥤", "Soda en Lata", 35, 1.00),
            new Producto(12, "🌭", "Hot Dog Clásico", 15, 2.00)
        ];
        guardarInventario();
    }
}
```

En esta clase de JavaScript se tiene todos los contenidos necesarios también para cargar las facturas, almacenar un historial de los clientes y una función para dar por finalizada la compra en donde se actualiza el stock de los productos, se asigna un subtotal, los impuestos y el total final de la compra:

```
function cargarFactura() {
    const tablaBody = document.getElementById("tablaProductosBody");
    const contenidoFactura = document.getElementById("contenido-factura");
    const mensajeVacio = document.getElementById("mensaje-vacio");
    const btnFinalizar = document.getElementById("btnFinalizar");

    // Validar si el carrito está vacío
    if (carritoCompra.length === 0) {
        contenidoFactura.style.display = "none";
        mensajeVacio.style.display = "block";
        btnFinalizar.disabled = true;
        btnFinalizar.style.opacity = "0.5";
        btnFinalizar.style.cursor = "not-allowed";
        return;
    }

    document.getElementById("fechaFactura").textContent = new Date().toLocaleDateString();
    document.getElementById("numTransaccion").textContent = "TX-" + Math.floor(Math.random() * 1000000);

    let subtotalGeneral = 0;
    tablaBody.innerHTML = "";

    // Construir las filas de la tabla
    carritoCompra.forEach(producto => {
        const totalLinea = producto.precio * producto.cantidad;
        subtotalGeneral += totalLinea;

        const fila = `
            <tr>
                <td>${producto.nombre}</td>
                <td class="texto-derecha">${producto.cantidad}</td>
                <td class="texto-derecha">$${producto.precio.toFixed(2)}</td>
                <td class="texto-derecha">$${totalLinea.toFixed(2)}</td>
            </tr>
        `;
        tablaBody.innerHTML += fila;
    });

    // Calcular impuestos y total final
    const impuestos = subtotalGeneral * TASA_IMPUESTO;
    const totalPagar = subtotalGeneral + impuestos;

    // Actualizar el DOM con los nuevos IDs en español
    document.getElementById("montoSubtotal").textContent = "$" + subtotalGeneral.toFixed(2);
    document.getElementById("montoImpuesto").textContent = "$" + impuestos.toFixed(2);
    document.getElementById("montoTotalGeneral").textContent = "$" + totalPagar.toFixed(2);
}
```

```
function registrarHistorico(detalles) {
    const hist = JSON.parse(localStorage.getItem('historico')) || [];
    hist.push(detalles);
    localStorage.setItem('historico', JSON.stringify(hist));
}

function finalizarCompra() {
    if (carritoCompra.length === 0) return;

    if (confirm("¿Confirmar compra y finalizar?")) {
        // calcular montos nuevamente para guardar en historial
        let subtotal = 0;
        carritoCompra.forEach(p => subtotal += p.precio * p.cantidad);
        const impuestos = subtotal * TASA_IMPUESTO;
        const totalPagar = subtotal + impuestos;

        const detalles = {
            fecha: new Date().toLocaleString(),
            transaccion: document.getElementById("numTransaccion").textContent,
            items: carritoCompra,
            subtotal: subtotal,
            impuestos: impuestos,
            total: totalPagar
        };
        registrarHistorico(detalles);

        localStorage.removeItem("carrito");
        alert("¡Compra realizada con éxito! Gracias por su preferencia.");
        window.location.href = "index.html";
    }
}
```

Ahora tenemos los archivos (o interfaces) en la raíz del repositorio los cuales son “carrito.html”, “factura.html”, historico.html”, “index.html”. dentro de estos archivos únicamente mandamos a llamar las funciones que previamente creamos en el archivo .js para poder dar su utilización.

Funciones utilizadas en el archivo “carrito.html” en donde se puede apreciar como se muestra el carrito, como se guarda el inventario, eliminan productos ya comprados y se actualiza su cantidad:

```
function cargarInventario() {
    const datos = localStorage.getItem('inventario');
    if (!datos) return [];
    return JSON.parse(datos);
}

function guardarInventario(inventario) {
    localStorage.setItem('inventario', JSON.stringify(inventario));
}

function ajustarStock(productoId, cantidad) {
    const inv = cargarInventario();
    const item = inv.find(p => p.id === productoId);
    if (item) {
        item.stock += cantidad; // cantidad puede ser negativa
        guardarInventario(inv);
    }
}

function mostrarCarrito() {
    const lista = document.getElementById("listaCarrito");
    const totalElemento = document.getElementById("total");

    lista.innerHTML = "";
    let total = 0;

    carrito.forEach((producto, index) => {
        const subtotal = producto.precio * producto.cantidad;
        total += subtotal;

        lista.innerHTML += `
            <div class="cart-item-row">
                <div class="item-info">
                    <h3>${producto.nombre}</h3>
                    <p>Precio U: $${producto.precio.toFixed(2)}</p>
                </div>
                <div class="item-controls">
                    <!-- cantidad editable -->
                    <input type="number" class="item-qty" value="${producto.cantidad}" min="1" data-index="${index}" onchange="actualizarCantidad(${index}, this.value)">
                    <span class="item-subtotal">$${subtotal.toFixed(2)}</span>
                    <button onclick="eliminarProducto(${index})" class="btn-delete">Eliminar</button>
                </div>
            </div>
        `;
    });

    totalElemento.textContent = "Total General: $" + total.toFixed(2);
}
```

```
function eliminarProducto(index) {
    // devolver el stock del articulo eliminado
    const producto = carrito[index];
    ajustarStock(producto.id, producto.cantidad);

    carrito.splice(index, 1);
    localStorage.setItem("carrito", JSON.stringify(carrito));
    mostrarCarrito();
}


// actualiza la cantidad de un producto en el carrito
function actualizarCantidad(index, cantidadNueva) {
    cantidadNueva = parseInt(cantidadNueva);
    if (isNaN(cantidadNueva) || cantidadNueva < 1) {
        // no se permiten valores inválidos ni menores a 1
        mostrarCarrito();
        return;
    }
```

Funciones utilizadas en el archivo “historico.html” en donde se puede apreciar cómo se cargan los historiales del cliente:

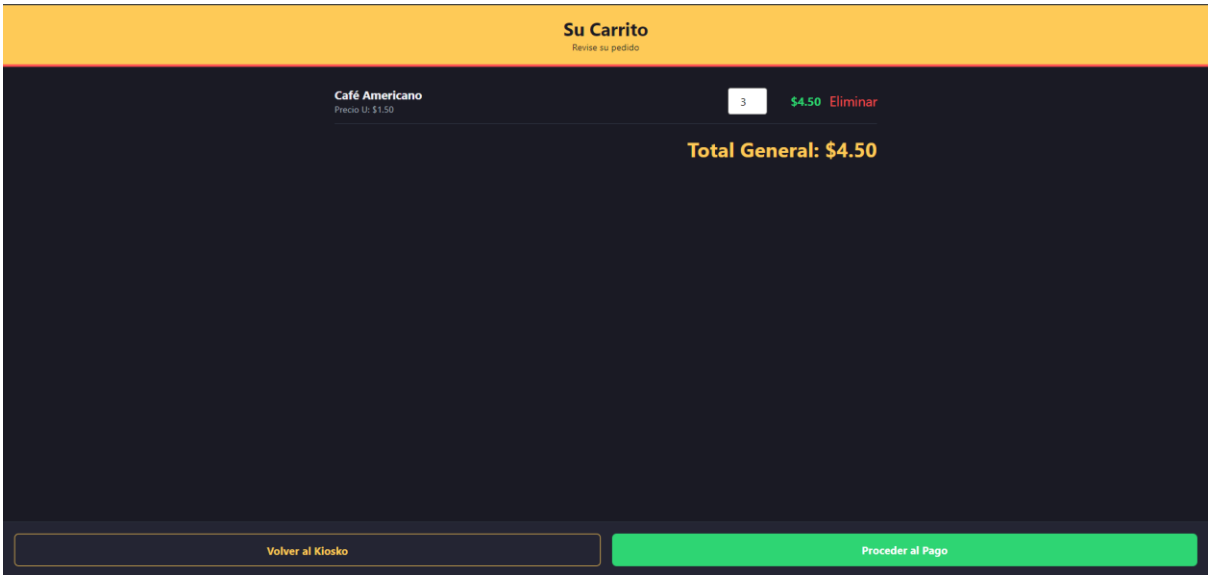
```
const listaHistorico = document.getElementById('listaHistorico');
const historico = JSON.parse(localStorage.getItem('historico')) || [];

function mostrarHistorico() {
    if (historico.length === 0) {
        listaHistorico.innerHTML = '<p>No hay registros de compras.</p>';
        return;
    }

    historico.forEach(entry => {
        const section = document.createElement('div');
        section.classList.add('caja-factura');
        let filas = '';
        entry.items.forEach(p => {
            filas += `
                <tr>
                    <td>${p.nombre}</td>
                    <td class="texto-derecha">${p.cantidad}</td>
                    <td class="texto-derecha">$${p.precio.toFixed(2)}</td>
                    <td class="texto-derecha">$${(p.precio * p.cantidad).toFixed(2)}</td>
                </tr>
            `;
        });

        section.innerHTML = `
            <div class="detalles-encabezado-factura">
                <div><strong>Fecha:</strong> ${entry.fecha}</div>
                <div><strong>No. Transacción:</strong> ${entry.transaccion}</div>
            </div>
            <table class="tabla-factura">
                <thead>
                    <tr>
                        <th>Producto</th>
                        <th class="texto-derecha">Cant.</th>
                        <th class="texto-derecha">Precio U.</th>
                        <th class="texto-derecha">Total Linea</th>
                    </tr>
                </thead>
                <tbody>
                    ${filas}
                </tbody>
            </table>
            <div class="area-totales-factura">
                <table class="tabla-totales">
```

Como ejecutar la aplicación:

Utilizando el sitio online:

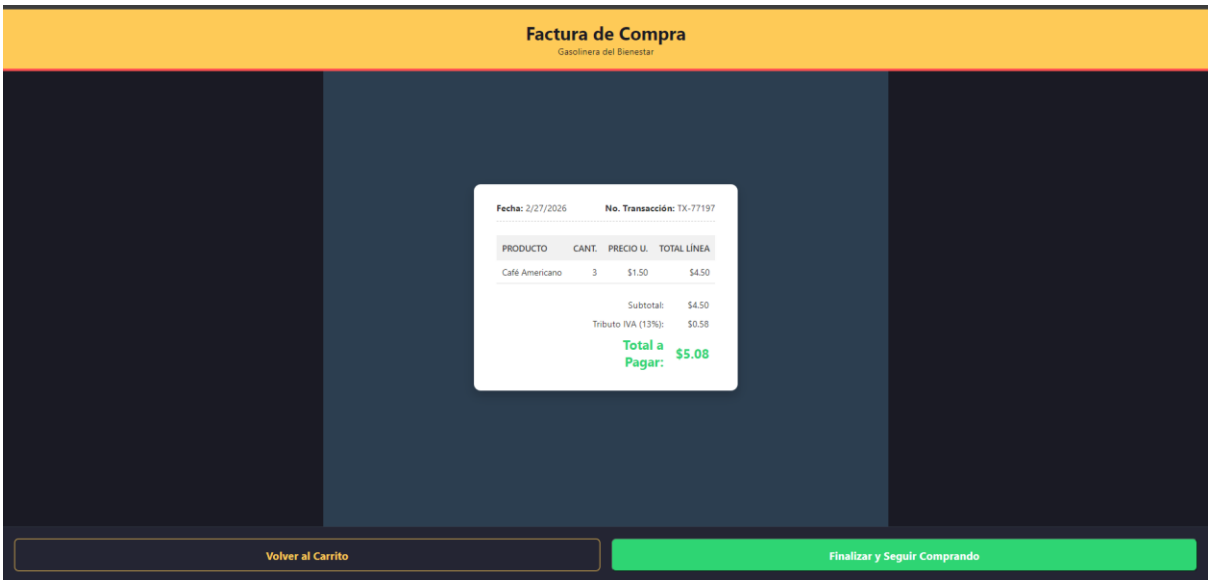
1. Debe de ingresar al siguiente link: [Kiosko Del Bienestar]
2. Luego ingresará directamente a la interfaz de los productos disponibles del kiosko, usted deberá de elegir la cantidad de los productos que desea y dar click al botón “Agregar” para agregarlo al carrito:

3. Luego deberá de bajar hasta encontrar el botón de “Ver carrito” para proceder a la siguiente interfaz:



Aquí podrá visualizar todos los productos que se agregaron a su carrito, la cantidad que lleva actualmente, el botón eliminar por si ya no desea dicho producto.

4. Para avanzar con la compra deberá de darle click al botón “Proceder el pago”.

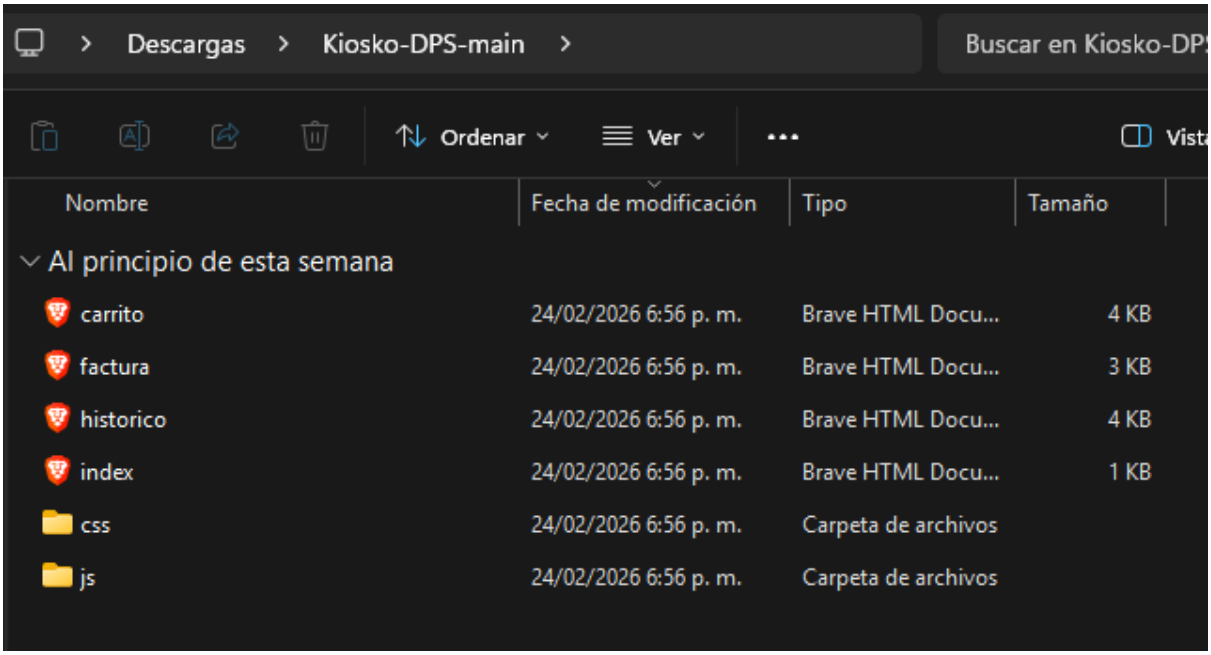


Usted podrá visualizar su factura detallada con los impuestos aplicados, la cantidad de los productos que lleva y el total a pagar.

5. Una vez finalizada su compra regresará a la interfaz de productos y terminará su procedimiento.

Utilizando el repositorio de GitHub de manera local:

1. Deberá ingresar al link del repositorio: [Repositorio del Kiosko Del Bienestar]
2. Una vez dentro del repositorio de GitHub deberá de darle click al botón "<> Code" y apretar el botón que diga "Descargar como ZIP", luego de su descarga deberá de descomprimirlo y tendrá lo siguiente:



3. Una vez ahí simplemente tendrá que dar doble click al archivo "index" y lo llevará directamente a la interfaz de compra de productos del Kiosko Del Bienestar y procede a realizar los pasos previamente explicados para el apartado de "Utilizando el sitio online".