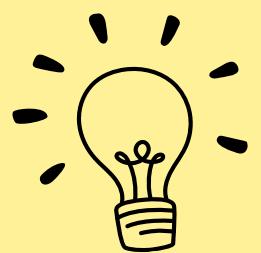




DESARROLLO DE UNA APP DE COMPRAS ONLINE: FUNCIONALIDADES Y DISEÑO



LIDIA GARCÍA MUÑOZ



FEB 2025

CONTENIDO

1. INTRODUCCIÓN

2. TECNOLOGÍAS UTILIZADAS

3. ESTRUCTURA CON HTML

4. FUNCIONALIDADES

5. CONCLUSIÓN



DESARROLLO WEB ENTORNO CLIENTE

I. INTRODUCCIÓN

Esta aplicación es un supermercado online diseñado para ofrecer una experiencia de compra sencilla y rápida. Los usuarios pueden explorar productos organizados en categorías, agregar artículos al carrito de compras y, finalmente, realizar la compra con un resumen detallado.

En esta presentación, analizaremos la estructura del código, las tecnologías utilizadas y cómo se implementan sus principales funcionalidades:



- Exploración de productos por categorías (frutas, carnes y pescados).
- Añadir productos al carrito con control de cantidades.
- Cálculo automático del subtotal y total de la compra.
- Resumen detallado de la compra en una ventana modal.
- Posibilidad de eliminar productos antes de finalizar la compra.

2. TECNOLOGÍAS UTILIZADAS



HTML

Lenguaje de marcado estándar utilizado para estructurar y presentar contenido en la web a través de un conjunto de etiquetas y atributos.



BOOTSTRAP

Framework de CSS que facilita la creación de sitios web responsivos y modernos con un sistema de grid, componentes pre-diseñados y plugins de JavaScript.



JAVASCRIPT

Lenguaje de programación que añade interactividad, permitiendo que los usuarios interactúen dinámicamente con los elementos de la página.



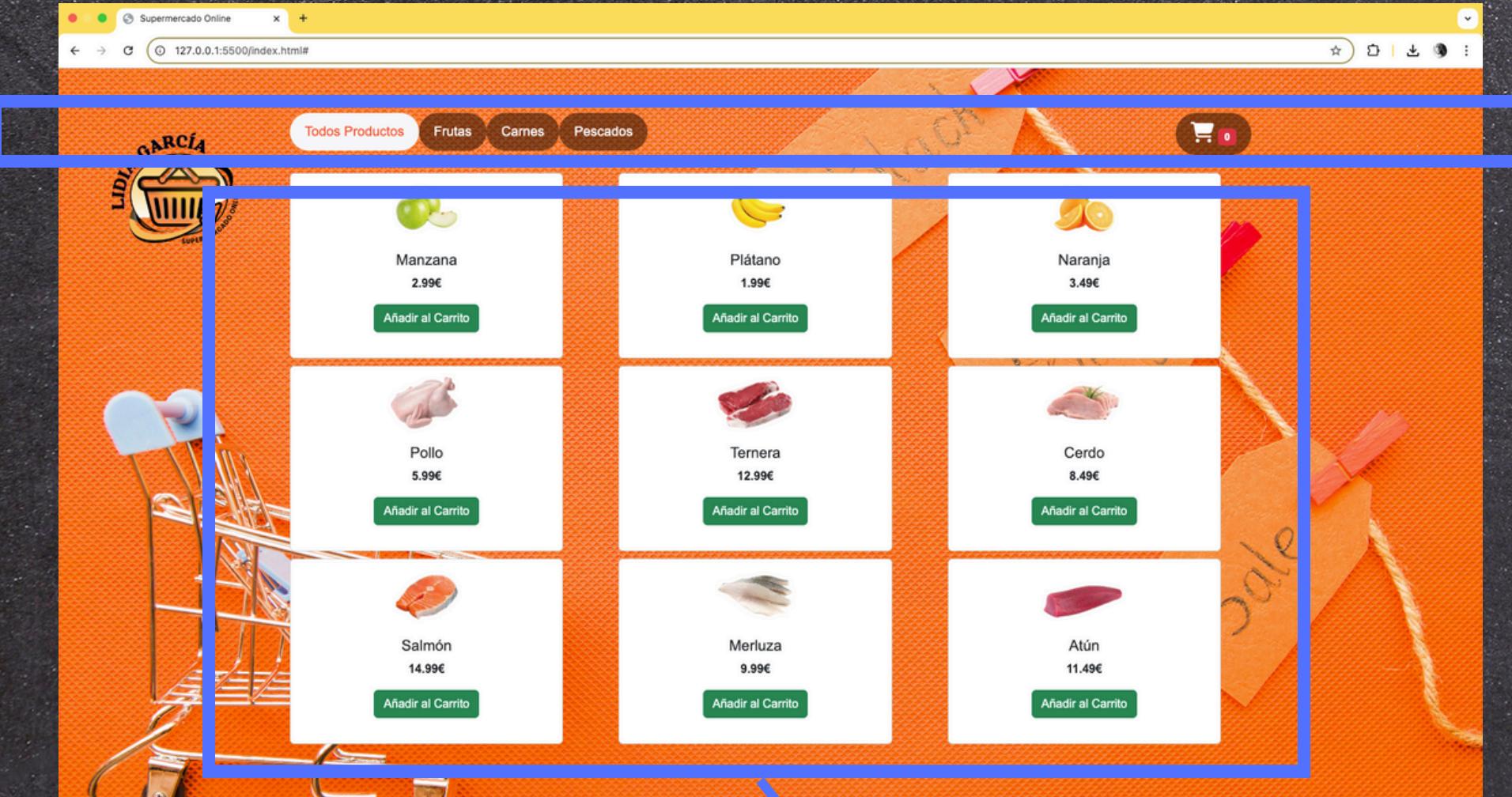
CSS

Lenguaje utilizado para describir la apariencia y el diseño de una página web. Se utiliza junto con HTML para controlar cómo se ven los elementos en la pantalla.

3. ESTRUCTURA CON HTML

La estructura HTML está organizada en secciones principales: barra de navegación, lista de productos, y modales para el carrito de compras y la finalización de la compra.

- Pestaña de navegación: permiten al usuario elegir entre diferentes categorías de productos (Todos los productos, Frutas, Carnes, Pescados, etc.)
- Sección de productos: muestra una lista de productos en formato de tarjetas.
- Modal de carrito: Se activa cuando el usuario desea ver el carrito de compras y finalizar la compra.
- Modal de compra: Muestra el resumen detallado de la compra una vez finalizada.

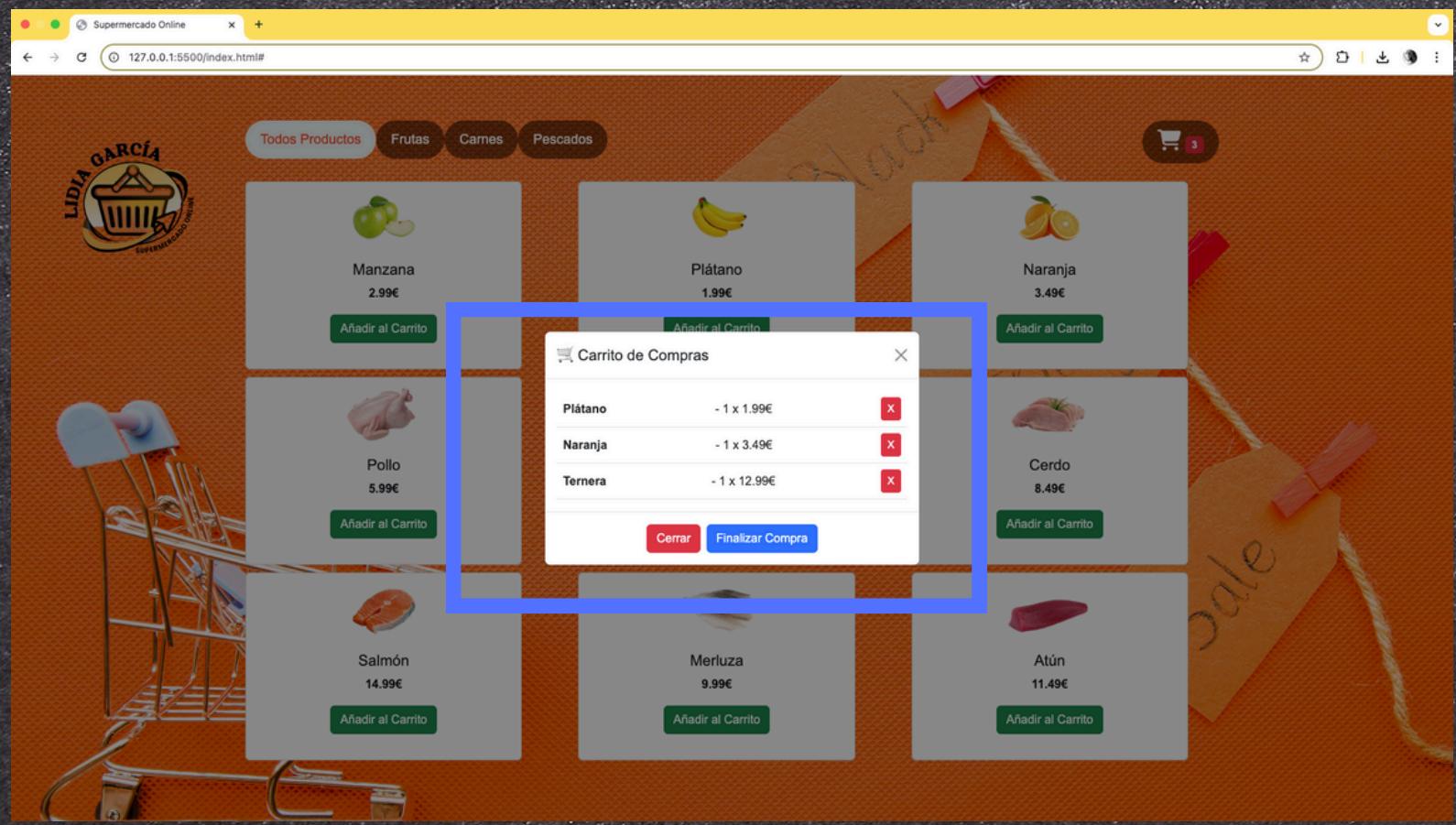


Sección de productos

```
<div id="productos" class="row mb-4">
    <!-- Aquí se mostrarán los productos -->
</div>
</div>
```

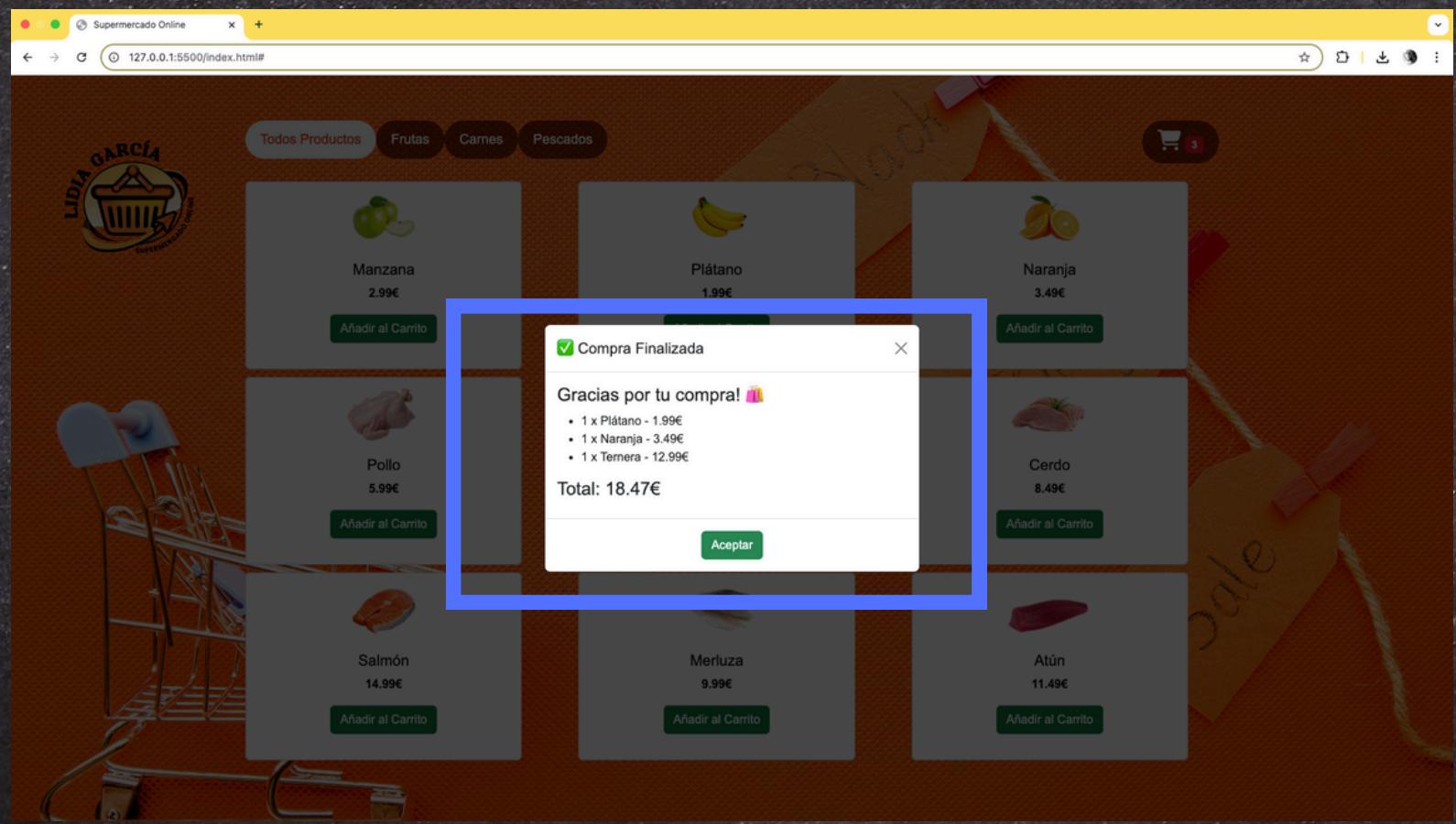
Pestaña de navegación

```
<!-- Barra de navegación con pestañas -->
<ul class="nav nav-pills mb-4">
    <li class="nav-item">
        <a class="nav-link active" id="todos-tab" href="#" onclick="mostrarProductos('todos')">Todos Productos</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" id="frutas-tab" href="#" onclick="mostrarProductos('frutas')">Frutas</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" id="carnes-tab" href="#" onclick="mostrarProductos('carnes')">Carnes</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" id="pescados-tab" href="#" onclick="mostrarProductos('pescados')">Pescados</a>
    </li>
    <!-- Carrito en el navbar a la derecha -->
    <li class="nav-item ms-auto">
        <a class="nav-link" href="#" data-bs-toggle="modal" data-bs-target="#modalCarrito">
            <i class="fas fa-shopping-cart carrito-icon"></i>
            <span id="contador-carrito" class="badge bg-danger">0</span>
        </a>
    </li>
</ul>
```



Modal de carrito

```
<div class="modal fade" id="modalCarrito" tabindex="-1">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">🛒 Carrito de Compras</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <div class="modal-body" id="lista-carrito"></div>
      <div class="modal-footer">
        <button class="btn btn-danger" data-bs-dismiss="modal">Cerrar</button>
        <button class="btn btn-primary" onclick="finalizarCompra()">Finalizar Compra</button>
      </div>
    </div>
  </div>
</div>
```



Modal de compra

```
<div class="modal fade" id="modalCompra" tabindex="-1">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">✅ Compra Finalizada</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" onclick="reiniciarCompra()"></button>
      </div>
      <div class="modal-body" id="resumen-compra"></div>
      <div class="modal-footer">
        <button class="btn btn-success" data-bs-dismiss="modal" onclick="reiniciarCompra()>Aceptar</button>
      </div>
    </div>
  </div>
</div>
```

4. FUNCIONALIDADES

La interactividad permite que los usuarios participen activamente con los contenidos en lugar de ser solo espectadores. En esta plataforma ecommerce se han integrado elementos interactivos que mejoran la experiencia del usuario.

En las siguientes diapositivas veremos en profundidad las siguientes funcionalidades que permiten la interacción del usuario:



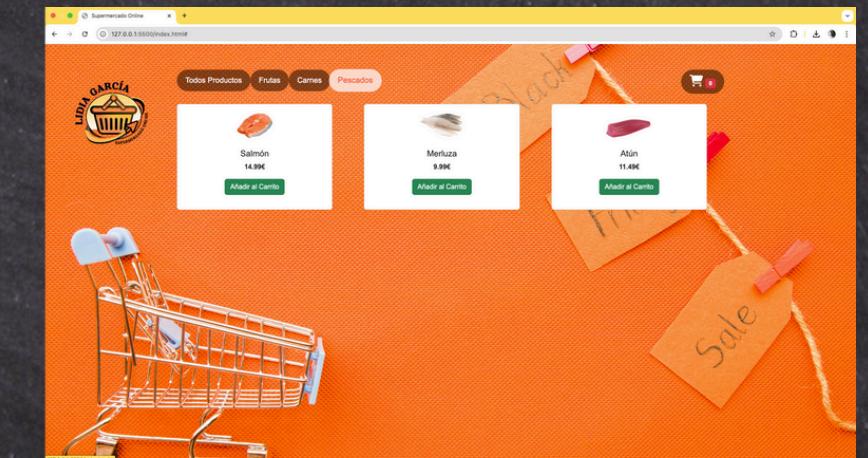
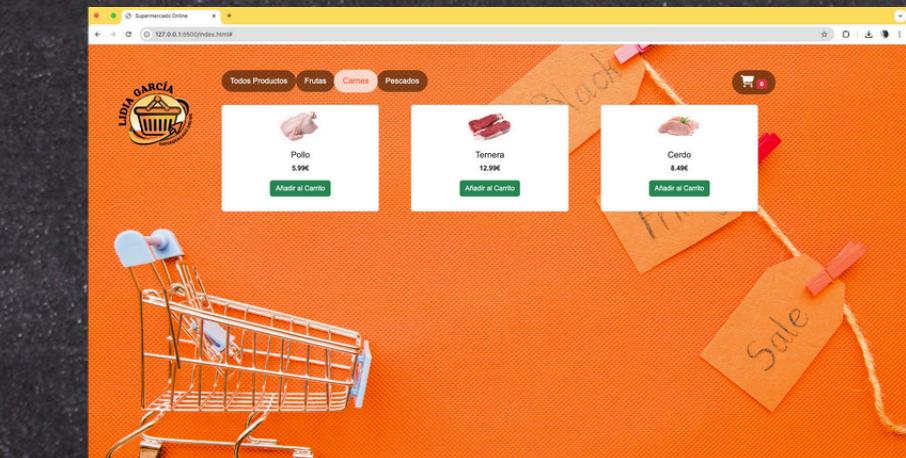
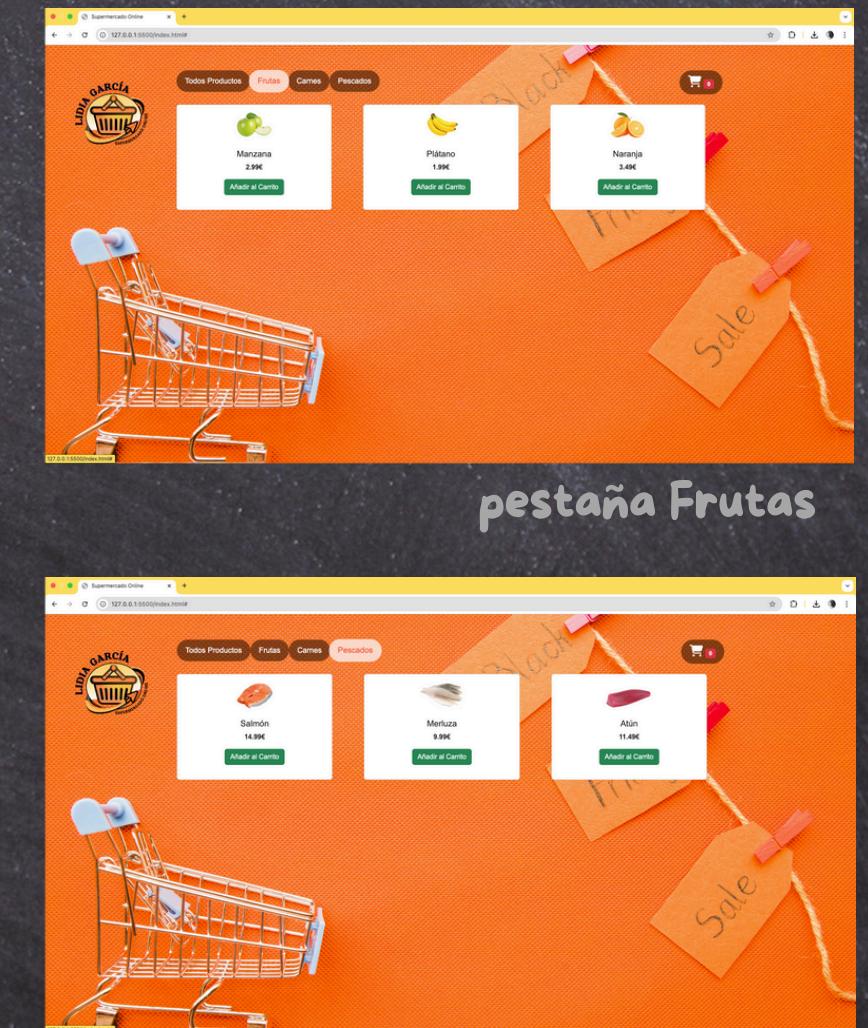
Mostrar productos

Los productos se muestran en tarjetas dentro de una cuadricula de 3 productos por fila en pantallas medianas.

Las categorías de productos se cargan dinámicamente en función de la selección del usuario (Todos los productos, Frutas, Carnes, Pescados,).

A continuación, parte del código JavaScript, para ver cómo se implementa esta funcionalidad :

```
function mostrarProductos(categoría) {  
    // Actualizar las pestañas  
    document.querySelectorAll('.nav-link').forEach(tab => {  
        tab.classList.remove('active');  
    });  
    if (categoría === 'todos') {  
        document.getElementById('todos-tab').classList.add('active');  
    } else {  
        document.getElementById(`#${categoría}-tab`).classList.add('active');  
    }  
  
    // Mostrar productos según la categoría  
    let productosMostrar = [];  
    if (categoría === 'todos') {  
        productosMostrar = [...productos.frutas, ...productos.carnes, ...productos.pescados];  
    } else {  
        productosMostrar = productos[categoría];  
    }  
  
    renderizarProductos(productosMostrar);  
}
```



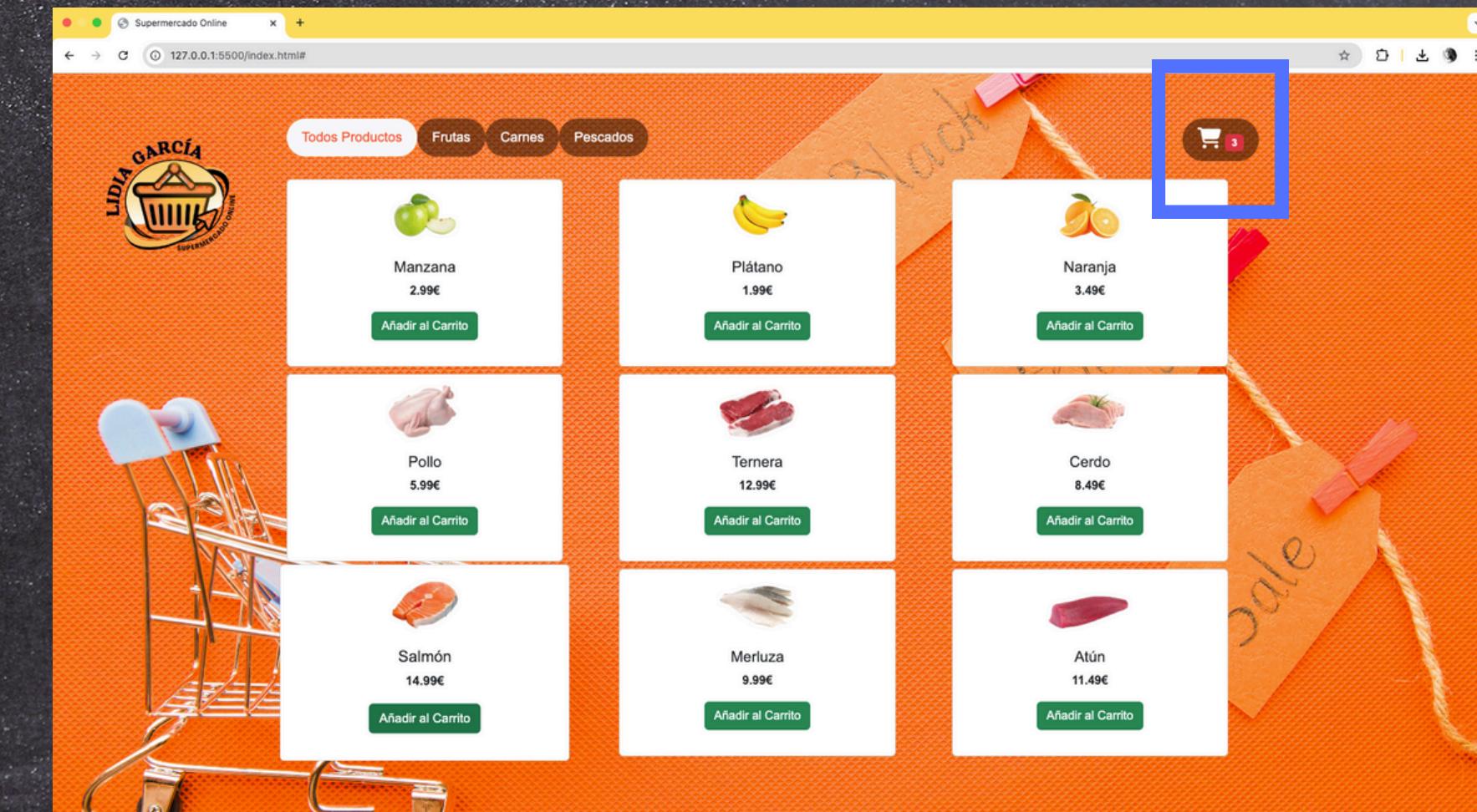
Agregar al carrito

Cada producto tiene un botón "Añadir al Carrito" que, al hacer clic, agrega el producto al carrito. Si el producto ya está en el carrito, se incrementa la cantidad y se actualiza el subtotal.

```
function agregarAlCarrito(id) {
  let producto;
  for (let categoria in productos) {
    producto = productos[categoria].find(p => p.id === id);
    if (producto) break;
  }

  let productoEnCarrito = carrito.find(p => p.id === id);
  if (productoEnCarrito) {
    productoEnCarrito.cantidad++;
    productoEnCarrito.subtotal = productoEnCarrito.precio * productoEnCarrito.cantidad;
  } else {
    carrito.push({ ...producto, cantidad: 1, subtotal: producto.precio });
  }

  actualizarCarrito();
}
```



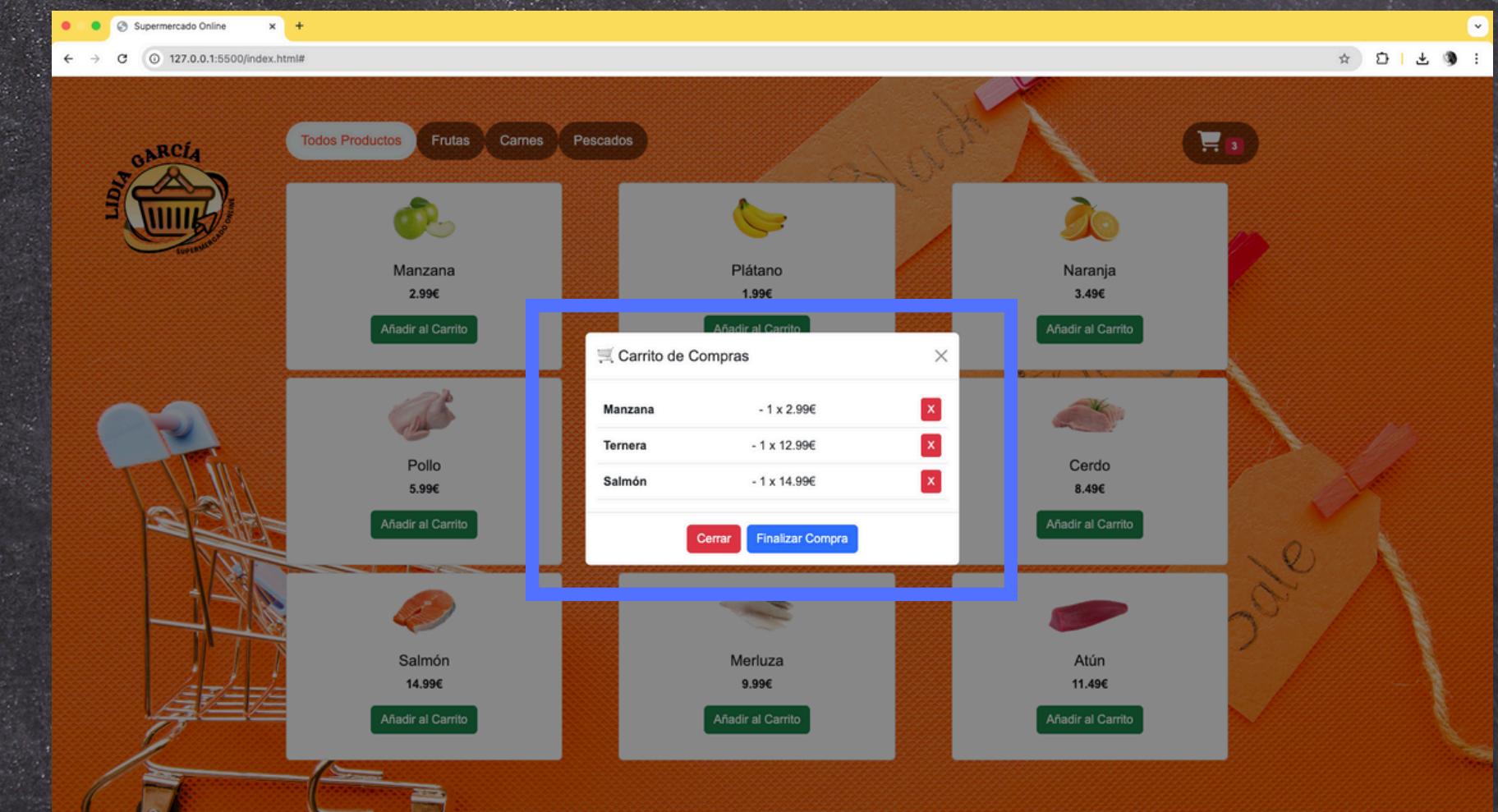
Mostrar el carrito

Actualiza la vista del carrito cada vez que se agregan o eliminan productos y muestra la cantidad de productos y sus respectivos subtotales. También actualiza el contador del carrito en el ícono de navegación.

```
function actualizarCarrito() {
  let lista = document.getElementById("lista-carrito");
  let contador = document.getElementById("contador-carrito");

  lista.innerHTML = "";
  contador.textContent = carrito.reduce((total, p) => total + p.cantidad, 0);

  carrito.forEach((p, index) => {
    lista.innerHTML += `
      <div class="d-flex justify-content-between align-items-center p-2 border-bottom">
        <strong>${p.nombre}</strong> - ${p.cantidad} x ${p.precio.toFixed(2)}€
        <button class="btn btn-danger btn-sm" onclick="eliminarProducto(${index})>X</button>
      </div>`;
  });
}
```

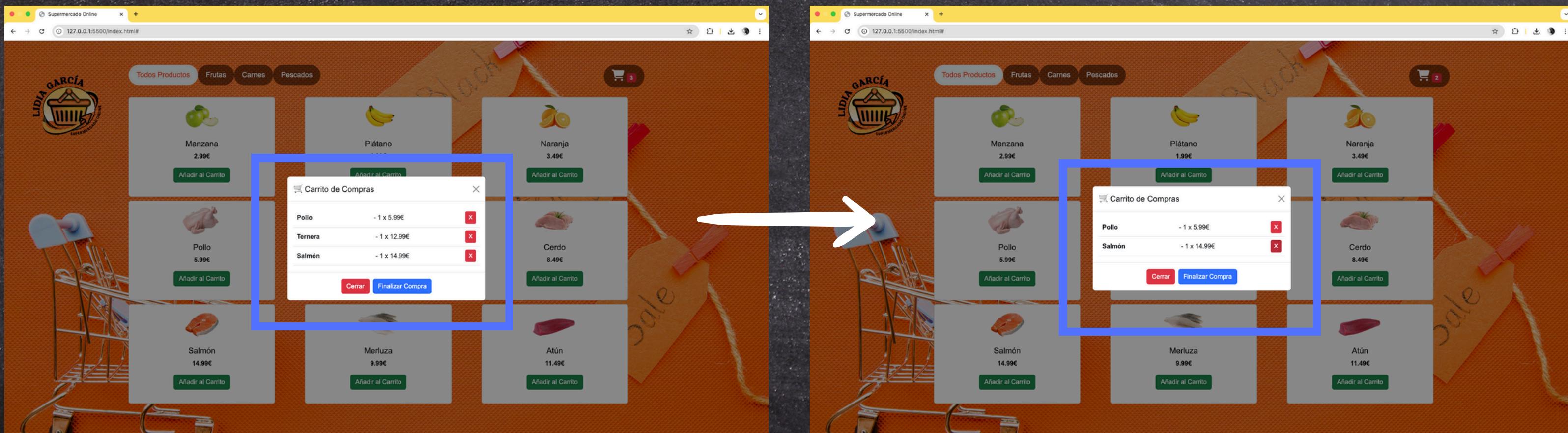


Eliminar productos del carrito

Permite a los usuarios **eliminar productos individualmente del carrito** antes de finalizar la compra. Al hacer clic en el botón de eliminar, el producto seleccionado se elimina del array del carrito y se actualiza la vista.

Si se eliminan todos los productos, el contador del carrito vuelve a cero.

```
function eliminarProducto(index) {  
    carrito.splice(index, 1);  
    actualizarCarrito();  
}
```



Finalizar. Resumen de la compra

Al hacer clic en "Finalizar Compra", se muestra un resumen de los productos en el carrito, con los subtotales de cada uno y el total general. Si el carrito está vacío, se muestra un mensaje de alerta.

```
function finalizarCompra() {
    if (carrito.length === 0) {
        alert("El carrito está vacío.");
        return;
    }

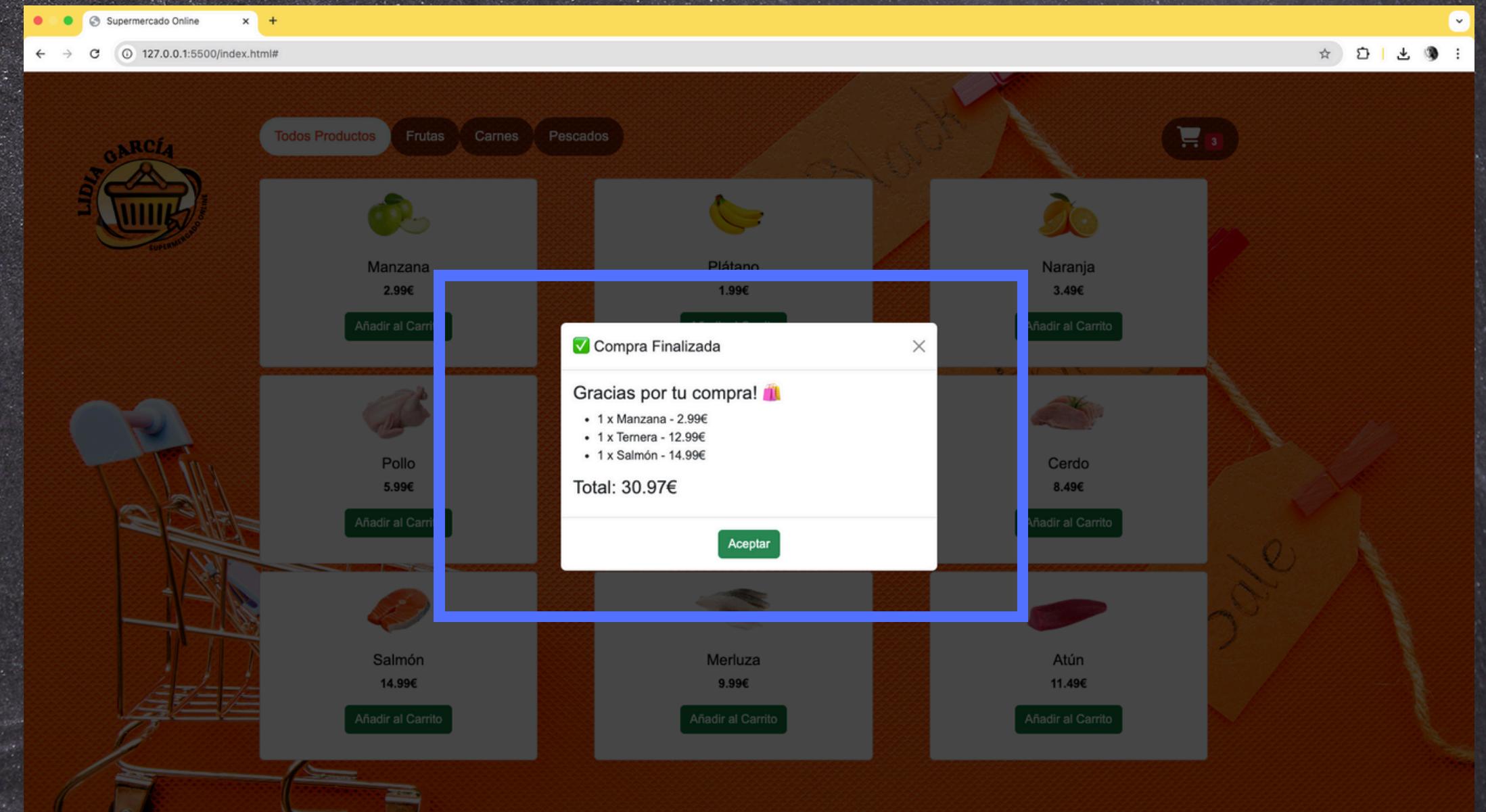
    let totalCompra = carrito.reduce((total, p) => total + p.subtotal, 0);
    let resumen = `<h4>Gracias por tu compra! 🎉</h4><ul>`;
    carrito.forEach(p => {
        resumen += `<li>${p.cantidad} x ${p.nombre} - ${p.subtotal.toFixed(2)}€</li>`;
    });
    resumen += `</ul><h4>Total: ${totalCompra.toFixed(2)}€</h4>`;

    document.getElementById("resumen-compra").innerHTML = resumen;
    new bootstrap.Modal(document.getElementById("modalCompra")).show();
}
```

127.0.0.1:5500 dice
El carrito está vacío.

Aceptar

mensaje de alerta



modal con resumen de la compra

6. CONCLUSIÓN

Esta aplicación demuestra cómo HTML, CSS y JavaScript pueden trabajar juntos para crear una experiencia de compra interactiva y fluida. Destaca la implementación de un carrito de compras dinámico, donde los usuarios pueden gestionar productos en tiempo real y obtener un resumen detallado antes de finalizar su compra.

El uso de Bootstrap garantiza un diseño responsive, optimizando la accesibilidad en distintos dispositivos. Además, la lógica en JavaScript permite una navegación sin recargas, mejorando la usabilidad.

Este proyecto es una base sólida para un e-commerce más avanzado, con posibles mejoras como autenticación de usuarios, pagos en línea y almacenamiento de pedidos. En definitiva, ofrece una experiencia intuitiva y eficiente, demostrando el potencial del desarrollo web en el comercio digital.

FUENTES

- ➡ Temario Cesur . Desarrollo Web Entorno Cliente
- ➡ Presentación Lidia García Muñoz - “Creación de una página web interactiva: HTML, CSS. y JavaScript en acción”

RECURSOS

- ➡ IDE Visual Studio Code
- ➡ Navegador Chrome

¡GRACIAS!



LIDIA GARCIA MUÑOZ

FEB 2025