Nombres: Daniel Diaz y Miguel Motta

**Trendify**

Trendify es una tienda en línea de ropa que permite a los usuarios registrarse e iniciar sesión para comprar sus productos favoritos utilizando el carrito de compras. Su interfaz amigable facilita la navegación por las distintas categorías de ropa y permite ver los detalles de cada producto. También cuenta con una barra de búsqueda y un carrito de compras, donde los usuarios pueden gestionar los productos que desean adquirir.

**Retrospectiva:**

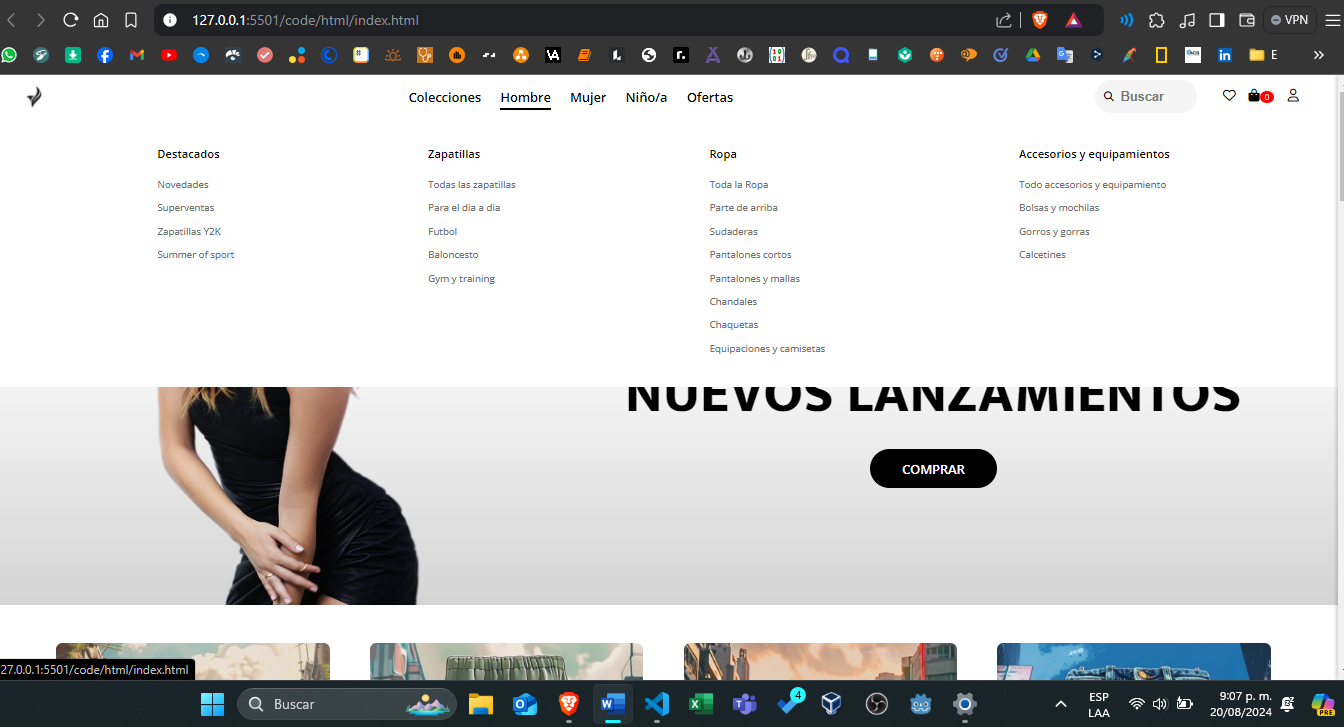
**Mejor implementación de código HTML y CSS:**

La sección del menú principal es donde hemos implementado la estructura HTML y el estilo CSS de manera más compleja y detallada. El código fuente de HTML está en el archivo index.html y los estilos CSS en los archivos Main.css y Home.css.

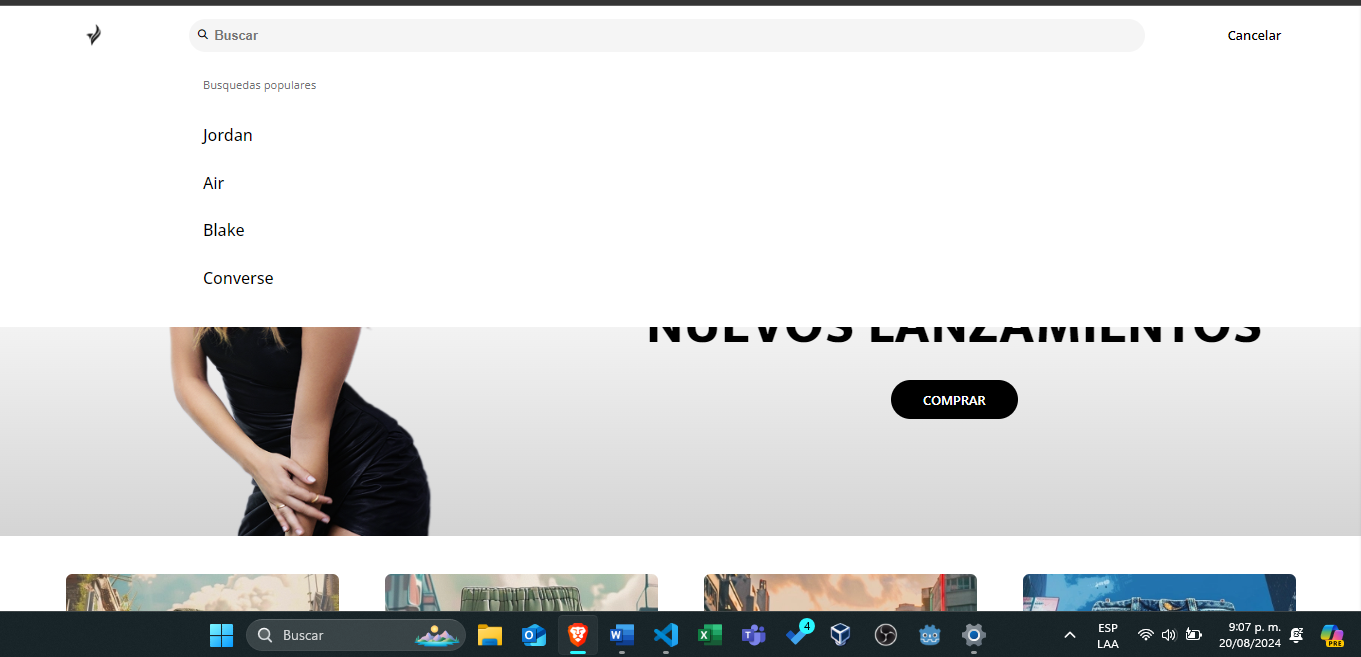
El menú superior es la sección más compleja. Se diseñó este menú de manera que sea responsive y se adapte a diferentes tamaños de pantalla. No solo se modifica su representación visual, sino también su estructura HTML, con el objetivo de brindar al usuario final una interfaz amigable independientemente del dispositivo que esté utilizando.

El documento index.html está estructurado de forma concreta, encapsulando los diferentes tipos de componentes en subtipos para mantener un código limpio y factorizable.

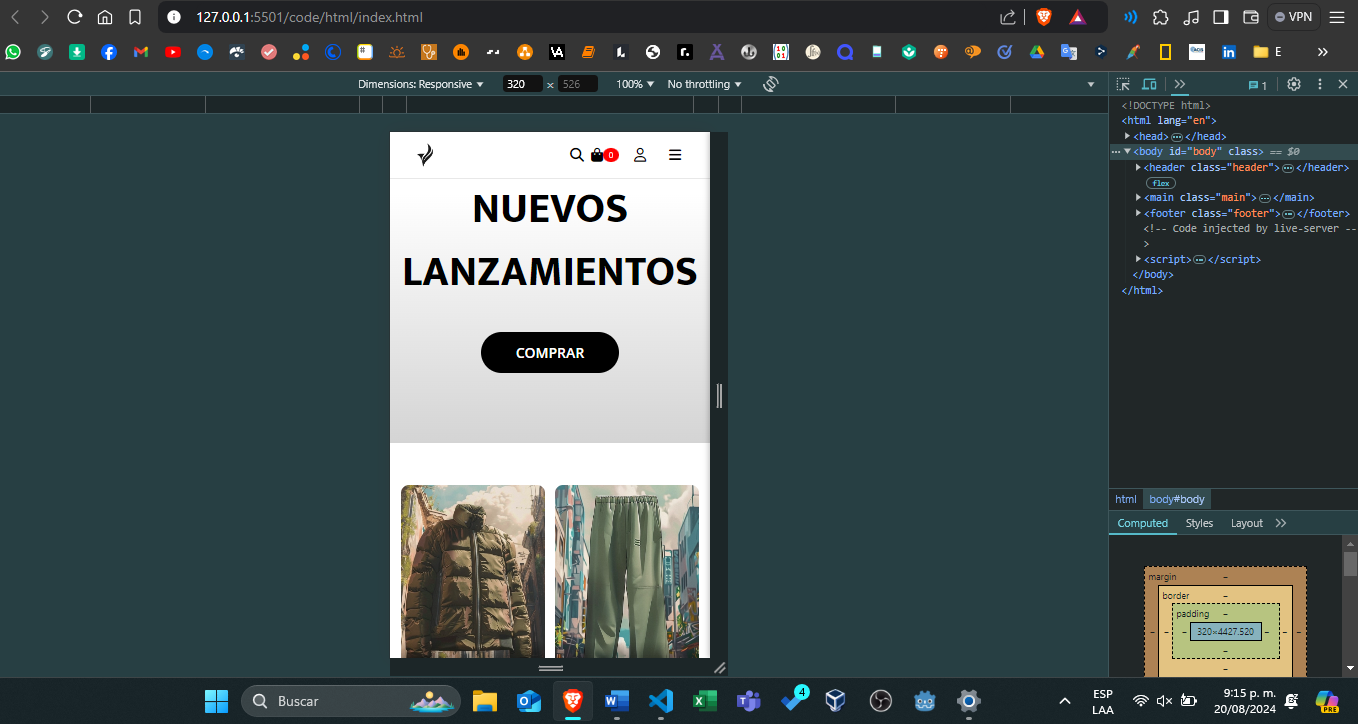
La vista del menú desplegable:

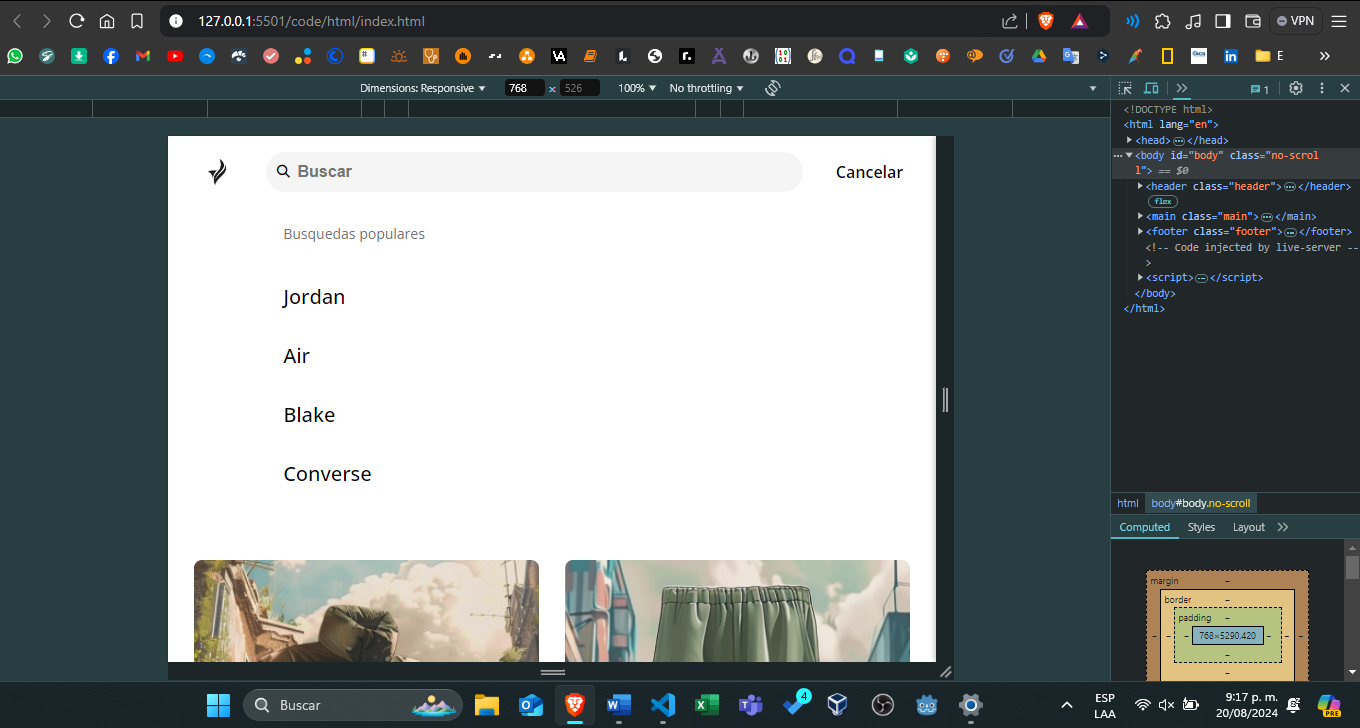


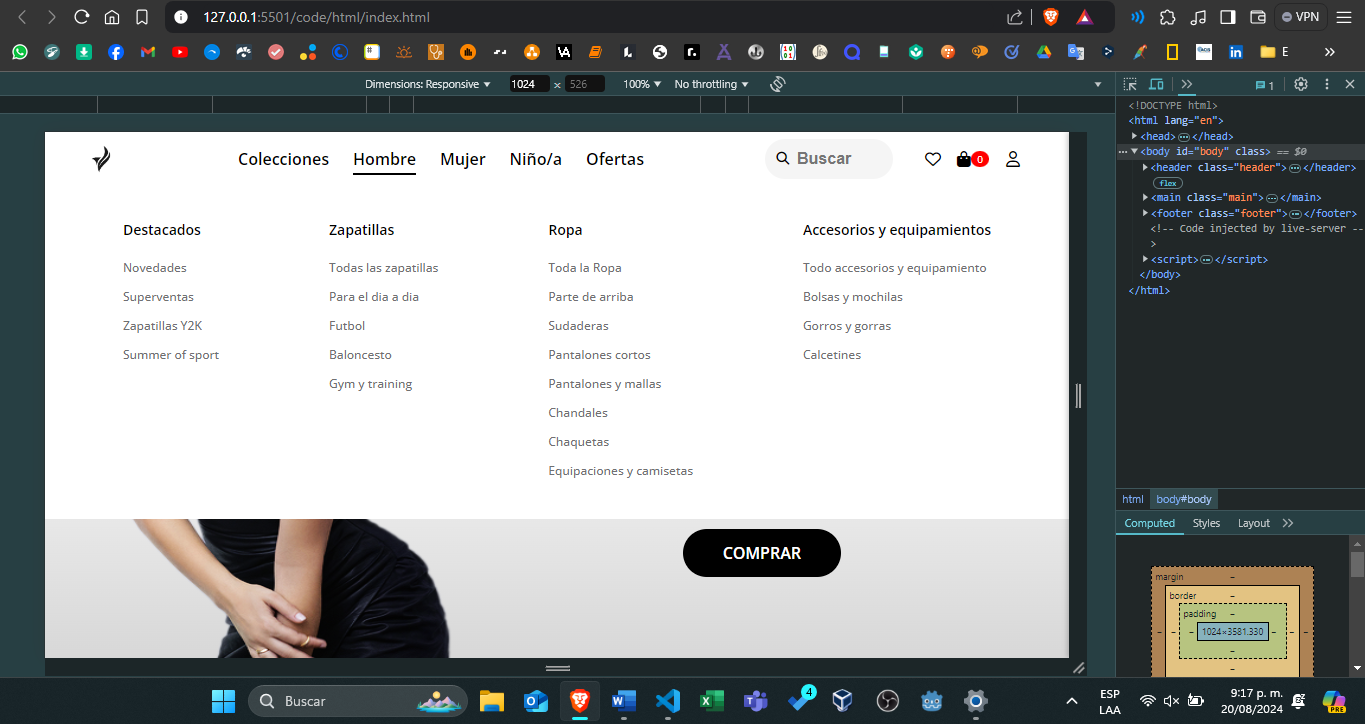
La vista de la barra desplegable de búsqueda:

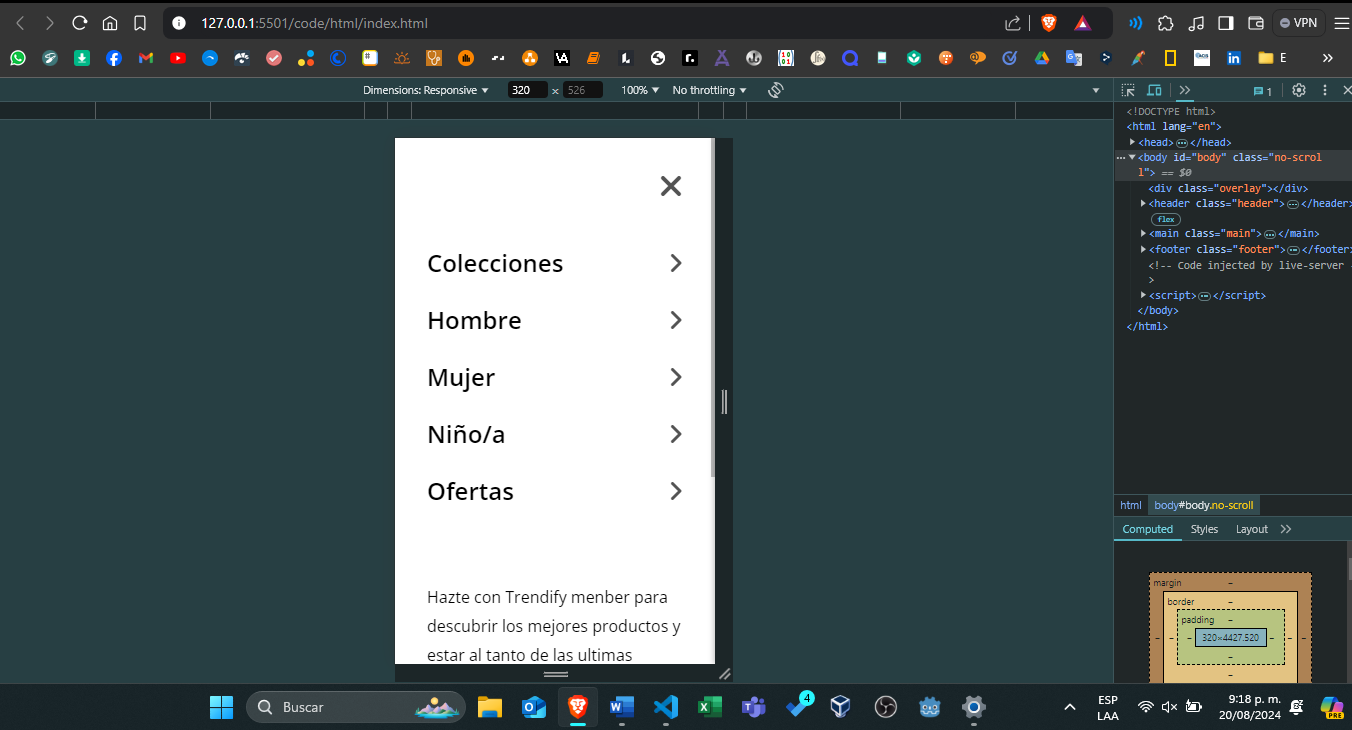
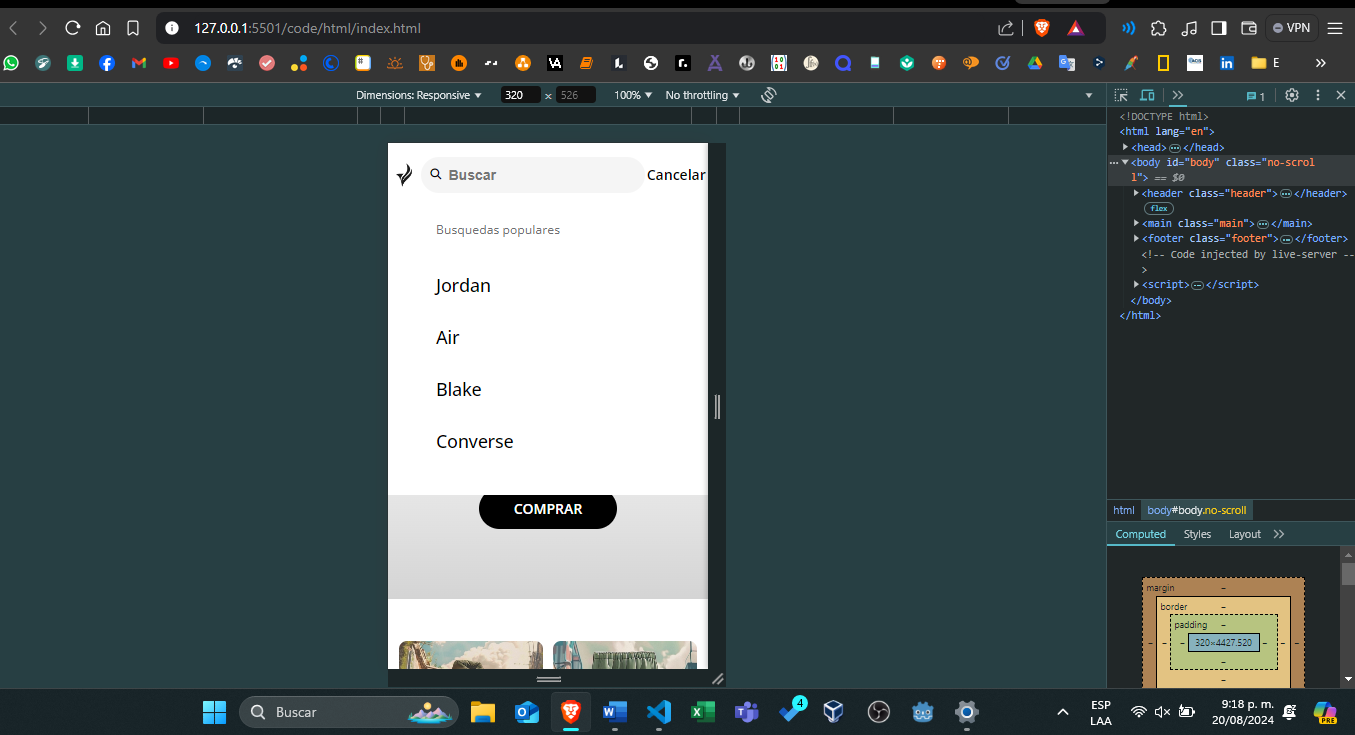


Vista de la aplicación expuesta a distintos tamaños de pantalla:







**Mejor implementación de JavaScript:**

En la sección del carrito de compras, hemos implementado un modelo MVC (model-view-controller). El modelo lógico se encarga de las operaciones relacionadas con el carrito de compras, como mantener un arreglo de productos, calcular los valores totales y subtotales, y gestionar la información del costo de envío y cantidad de cada producto. El modelo de vista se encarga del control de los componentes que se agregan, actualizan o eliminan en la interfaz de usuario, recibiendo los parámetros necesarios para crear sus respectivos componentes. Finalmente, el controlador contiene los métodos de interacción con el usuario y conecta el modelo lógico con la vista.

Además, se implementó un método para cargar el catálogo de productos disponibles en la página a través de un archivo CSV. Este archivo contiene la información de los productos y permite visualizarla desde un documento de Excel. Esta fue una solución para conectar una fuente de datos persistente con la interfaz de usuario, con el objetivo de mantener la información de los artículos independientemente del código fuente.

Podemos ver una sección del código de cart.js que muestra la implementación de un modelo MCV además de la implementación de la librería de JQuery:

