



ACTIVIDAD 1. DESARROLLO DEL DISEÑO RESPONSIVE DEL FRONT-END

A: DIANA MARCELA TOQUICA
INGENIERO DE SISTEMAS

DE: MAICOL STIVEN MORENO LÓPEZ

ASUNTO: INFORME TÉCNICO SOBRE LA
IMPLEMENTACIÓN Y EVALUACIÓN DEL
DISEÑO RESPONSIVE EN UN PROYECTO
WEB

FECHA: 09 DE FEBRERO DE 2025

Introducción

El propósito de este documento es explicar y resaltar la importancia de implementar la tecnología responsive en los proyectos web. La adaptabilidad de una página web a diferentes resoluciones y tamaños de pantalla es crucial para los desarrolladores que buscan ofrecer una experiencia de usuario fluida y agradable que permita a los internautas navegar por sus sitios web desde cualquier dispositivo sin importar cuán grande, pequeño, obsoleto o novedoso sea. Esto no solo amplía el alcance a más público, sino que también optimiza la experiencia de acuerdo a cada tipo de pantalla móvil.

En este trabajo, cuyo objetivo es crear y diseñar una página web simple enfocada al comercio en línea, se implementará la tecnología responsive utilizando las herramientas de las tecnologías web. Así, se evaluará si una buena página web se debe, en gran medida, a su capacidad de adaptarse a diversas resoluciones.

Objetivo

Evaluar el diseño responsive en proyectos web mediante el desarrollo de una página web adaptable a diferentes tamaños de pantalla, utilizando CSS Grid, Flexbox y media queries.

Desarrollo Y Pruebas

CSS Grid y Flexbox

Entre las herramientas disponibles en las tecnologías web, especialmente en CSS (Cascading Style Sheets), se destacan dos en particular: CSS Grid y Flexbox. Ambas están destinadas a gestionar la distribución de los elementos en una página web, permitiendo la creación de layouts únicos y organizando cada componente con la alineación, tamaño y espacio deseado. Estas herramientas son ideales para diseñar estructuras complejas y responsivas que se ajusten a diferentes tamaños de pantalla y dispositivos.

Puntualmente, CSS Grid se define como un “sistema de maquetación web que divide la página en una cuadrícula o rejilla (grid) a partir de la cual se pueden posicionar los diferentes elementos” (Mariño, 2023), mientras que Flexbox “permite centrar bloques vertical y horizontalmente en un contenedor y alinear estos elementos dinámicamente” (Lorca, 2024).

Para el proyecto web actual, se puso en práctica la herramienta Flexbox en algunos contenedores dentro de los archivos CSS de la página web de la tienda online. Gracias a la combinación de Flexbox con otros atributos, el diseño de la página permite una disposición y presentación del contenido de la tienda de manera ordenada y agradable. Esto evita la sobrecarga visual, respeta los espacios y posiciona correctamente cada elemento. A continuación, se presentarán algunas capturas de las líneas de código que utilizan estas herramientas en las siguientes Figuras, junto con su resultado final.

Figura 1.

Contenedores y elementos flexibles

```
.header-layout { /* Panel de Encabezado */
  display: flex;
  align-items: center;
  justify-content: space-between;
  letter-spacing: 1px;
  border-bottom: 2px solid #ffc000;
}
```



Nota. En la clase `header-layout`, un elemento HTML diseñado para contener y describir los elementos del encabezado de la página web, se activa el modelo Flexbox. Esto convierte al `header-layout` en un contenedor flexible, y todos los elementos dentro de él se transforman en elementos flexibles también, organizados de acuerdo a las especificaciones de las propiedades de Flexbox como son los atributos `align-items` y `justify-content`, los cuales puede distribuir horizontal o verticalmente los elementos según la alineación establecida por dichos atributos.

Figura 2.

Atributos de alineación, tamaño y espacio

```

124     flex-wrap: wrap;
125 }
126
127 .features-item { /* Panel de Producto */
128     display: flex;
129     flex-direction: column;
130     background-color: #fff;
131     padding: 20px;
132     border-radius: 10px;
133     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
134     width: 30%;
135     animation: fadeIn 1s ease-in-out;
136     flex-grow: 1; /* Hace que la descripción ocupe el espacio disponible */
137     justify-content: space-between; /* Distribuye el espacio para que el precio y botón estén abajo */
138 }
139

```



Nota. La línea de código CSS que se muestra define el estilo de los paneles de los artículos de la tienda, utilizando Flexbox para organizar los elementos en columnas (`flex-direction: column`). El diseño incluye un fondo blanco, relleno, bordes redondeados, una sombra suave, un ancho del 30% y una animación de aparición gradual. Los atributos `flex-grow` y `justify-content` distribuyen el espacio disponible del bloque a lo largo de la columna, haciendo que todos los elementos hijos aumenten de tamaño hasta ocupar el espacio disponible en el contenedor flexible.

Figura 3.

Solapamiento de contenedores

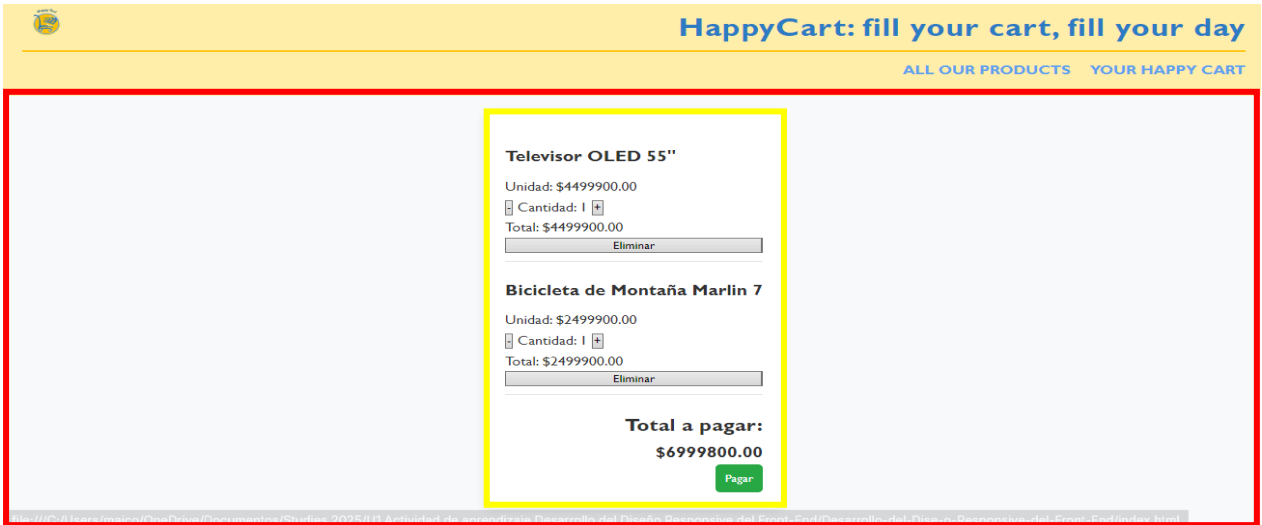
```

/* Sección de Compras */
main {
  flex: 1;
  max-width: 800px;
  margin: 20px auto;
  padding: 20px;
  background-color: white;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
  animation: fadeIn 1s ease-in-out;
}

/* Productos */
#cart-container { /* Contenedor del Carrito */
  margin-bottom: 20px;
}

.cart-item { /* Artículo del Carrito */
  display: flex;
  flex-direction: column;
  border-bottom: 1px solid #e0e0e0;
  padding: 10px 0;
  margin: 10px 0;
}

```



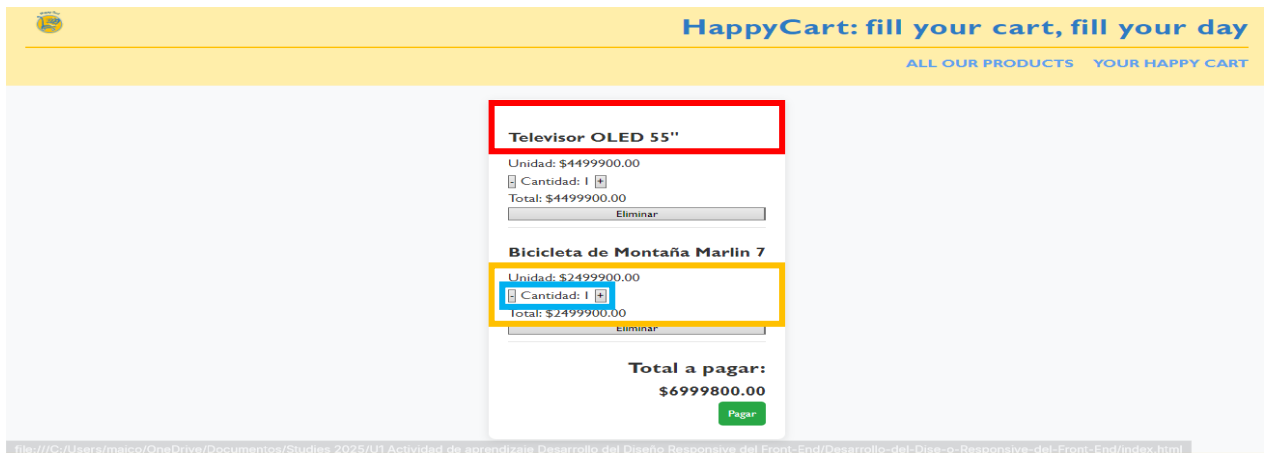
```

.item-details { /* Detalles del Artículo */
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 5px 0;
}

.item-price, .item-total { /* Precio Unitario y Total del Artículo */
  flex: 1;
  text-align: center;
}

.quantity-controls { /* Controles de Cantidad */
  display: flex;
  align-items: center;
}

```



Nota. Estas capturas de pantalla demuestran las capacidades de Flexbox para superponer bloques dentro de otros, ajustando la dirección y el sentido de los contenedores y sus elementos hijos. Utilizando los atributos mostrados, el contenido del carrito de compras como el precio unitario y total, los botones de cantidad y el botón de ‘Eliminar’, así como el nombre del producto, se pueden posicionar verticalmente, alineados en el centro del contenedor y con un espacio de 1px entre los ítems. Esto evidencia cómo se puede organizar el contenido de manera dinámica y flexible, justificando así el nombre de Flexbox.

Media Queries

Otra de las herramientas de la tecnología CSS disponible para los desarrolladores web son las Media Queries. Estas funciones “permiten especificar la visualización de un documento para diferentes medios de salida y tamaños de pantalla” (Seobility, s. f.), convirtiéndolas en una herramienta esencial para crear un diseño responsive óptimo. Con las Media Queries, no es necesario definir manualmente las resoluciones y tamaños de pantalla, ya que estas ajustan automáticamente las páginas web para ser visualizadas en varios dispositivos, tal como se muestra en la Figura 4.

Figura 4.

Adaptabilidad en tamaño en los medios móviles

```

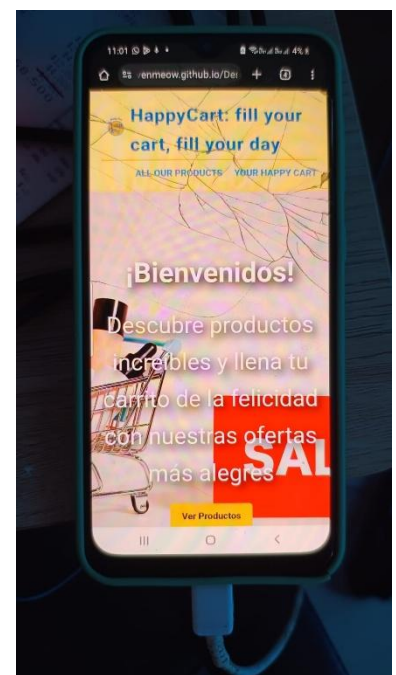
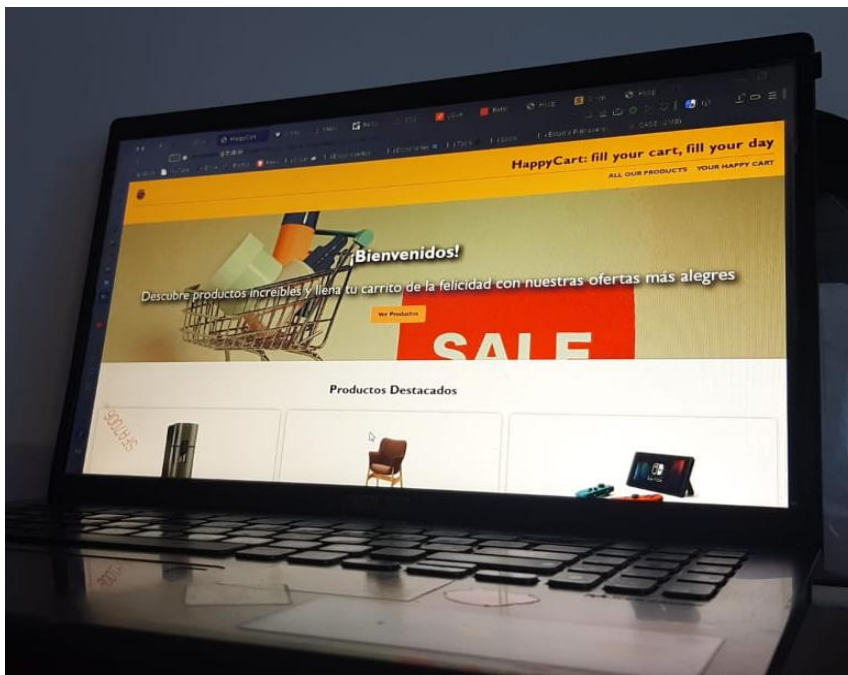
/* Responsive */
@media screen and (max-width: 768px) {
  .header h1 {
    font-size: 1.2rem;
  }

  nav ul {
    flex-direction: column; /* Cambia a columna en pantallas pequeñas */
  }

  nav ul li {
    margin: 10px 0; /* Espacio entre los elementos de la lista */
  }
}

@media screen and (max-width: 480px) {
  .header h1 {
    font-size: 1rem;
  }
}

```



Nota. El código CSS ajusta el tamaño de fuente del encabezado h1 y la disposición de la lista ul1 dentro de la navegación nav para diferentes tamaños de pantalla, específicamente para pantallas de hasta 768px y hasta 480px. Esto permite aplicar diferentes estilos al diseño de la página web basados en las características del dispositivo.

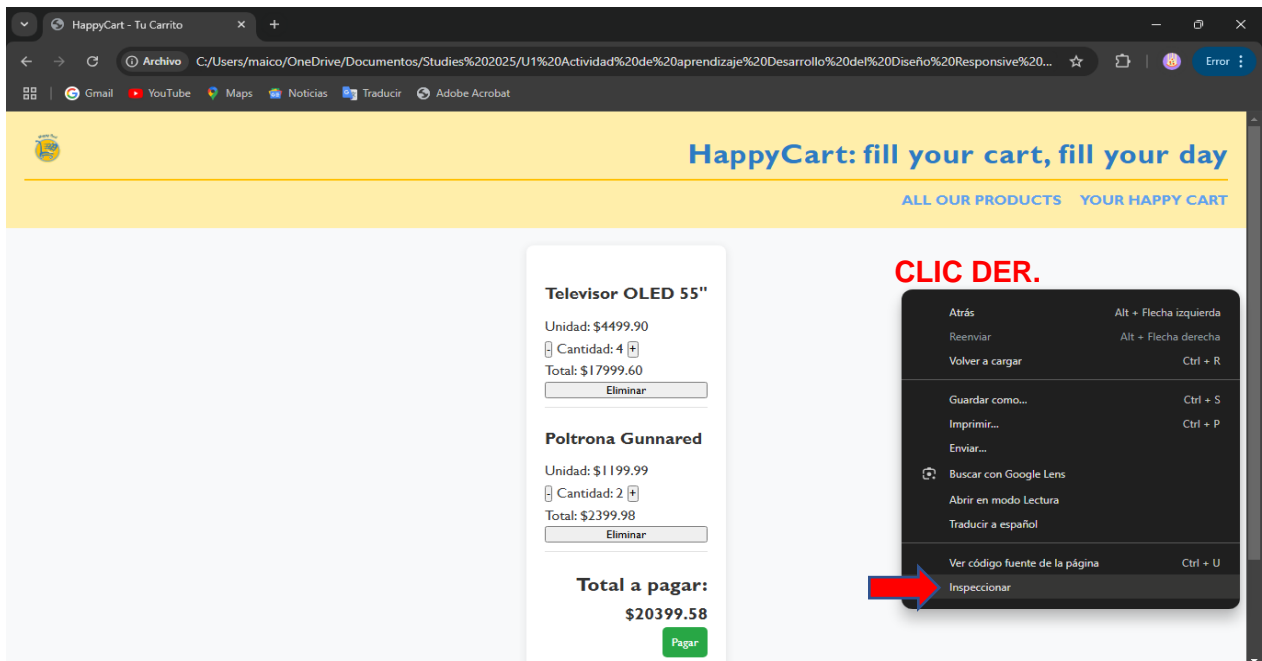
Chrome DevTools

De acuerdo a la definición de Google (s. f.) en su foro de ayuda, “Chrome DevTools consiste en un conjunto de herramientas web integrado en el navegador Chrome que permite a los desarrolladores inspeccionar el HTML renderizado (DOM) y la actividad de

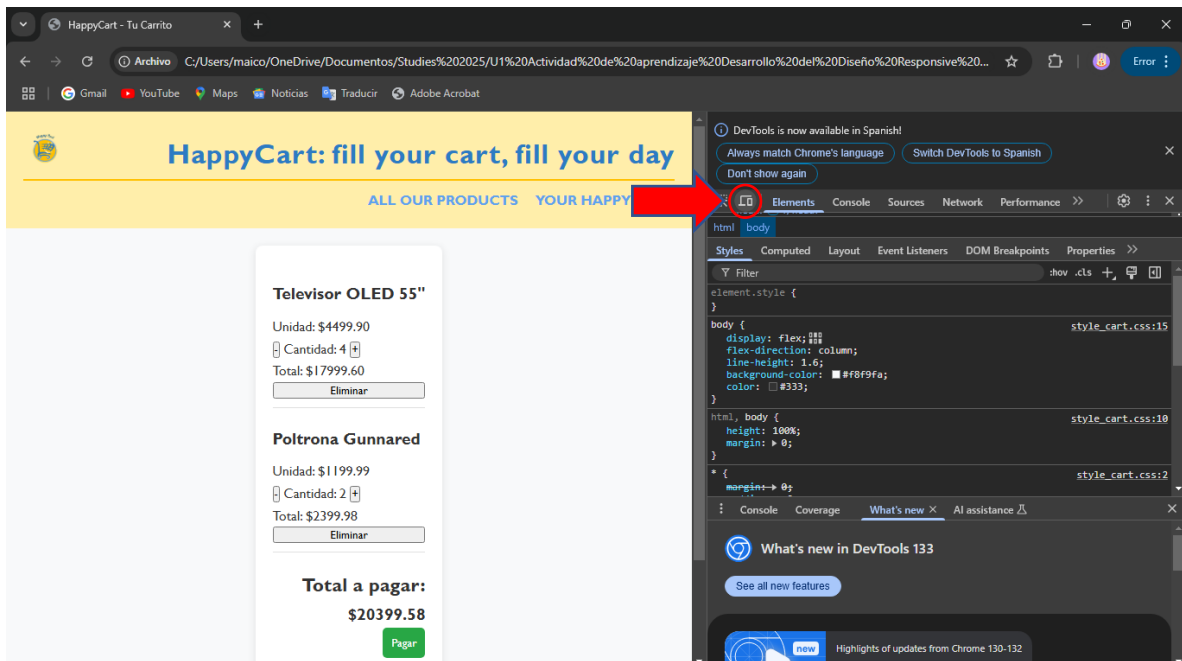
red de sus páginas.” Específicamente para probar que el diseño de la página web de la plataforma de comercio electrónico sea adaptable a diferentes dispositivos, se siguieron los siguientes pasos expuestos en la Figura 5.

Figura 5.

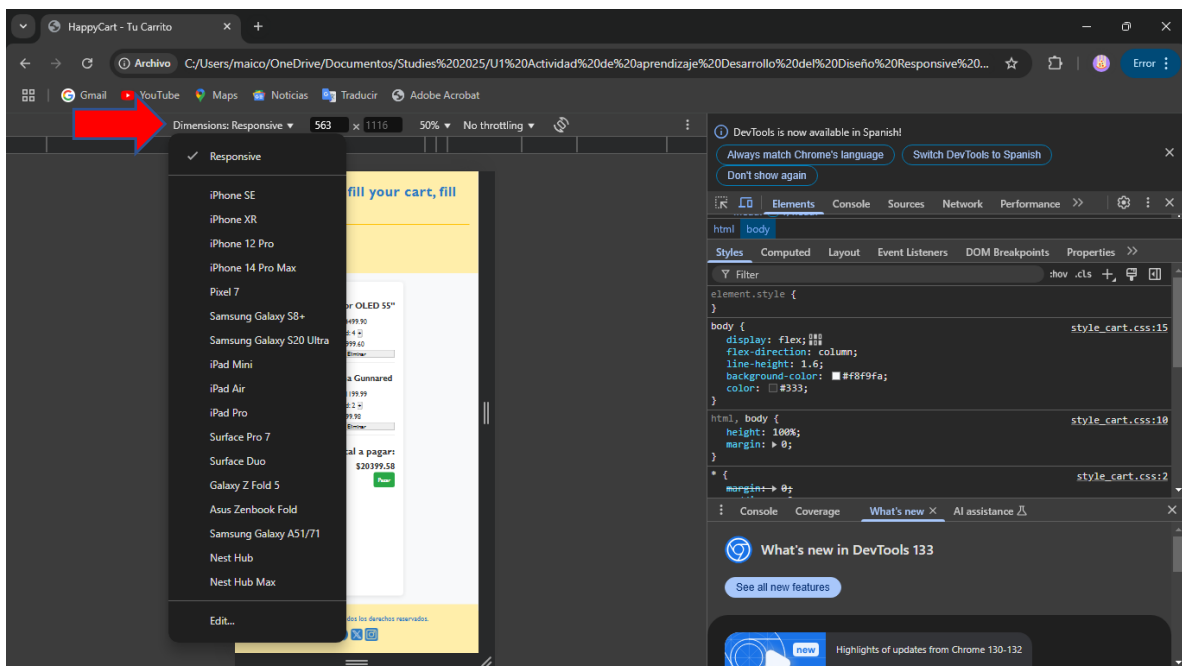
Pruebas responsive con Toggle Device Toolbar de Chrome DevTools



Nota. En tu navegador Chrome, accede a la página web de tu propiedad y haz clic derecho con el ratón sobre la pantalla. Aparecerá una ventana emergente. Luego, selecciona la opción 'Inspeccionar' al final del menú.

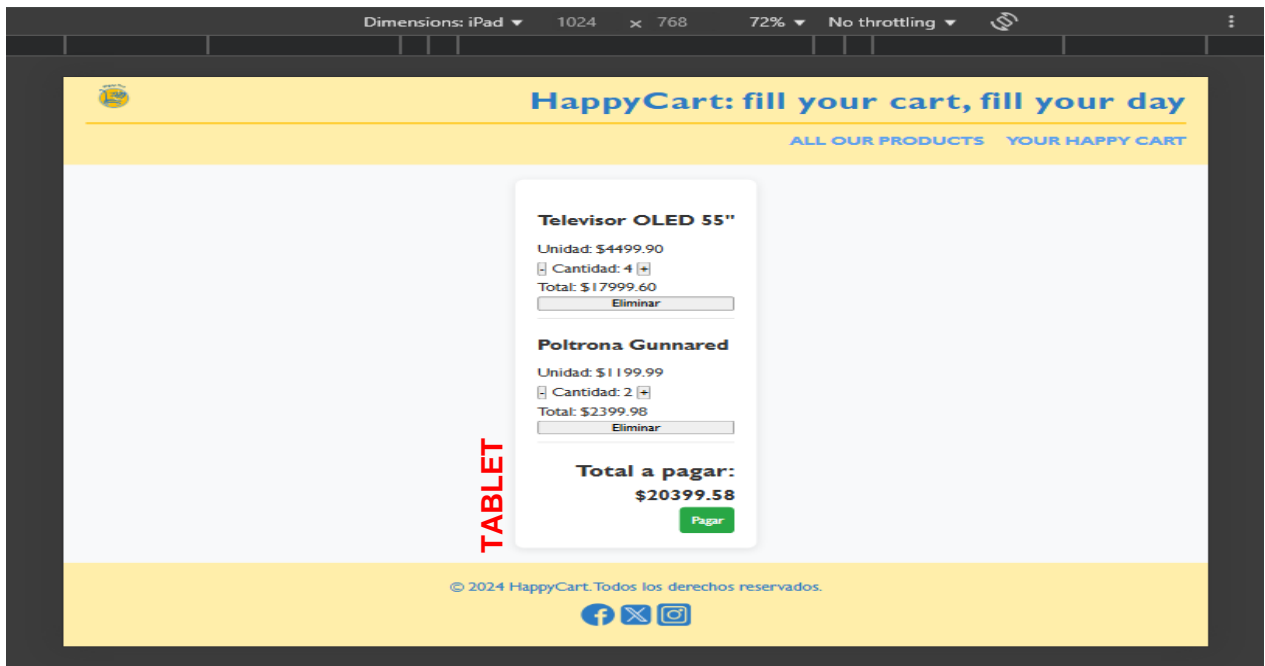


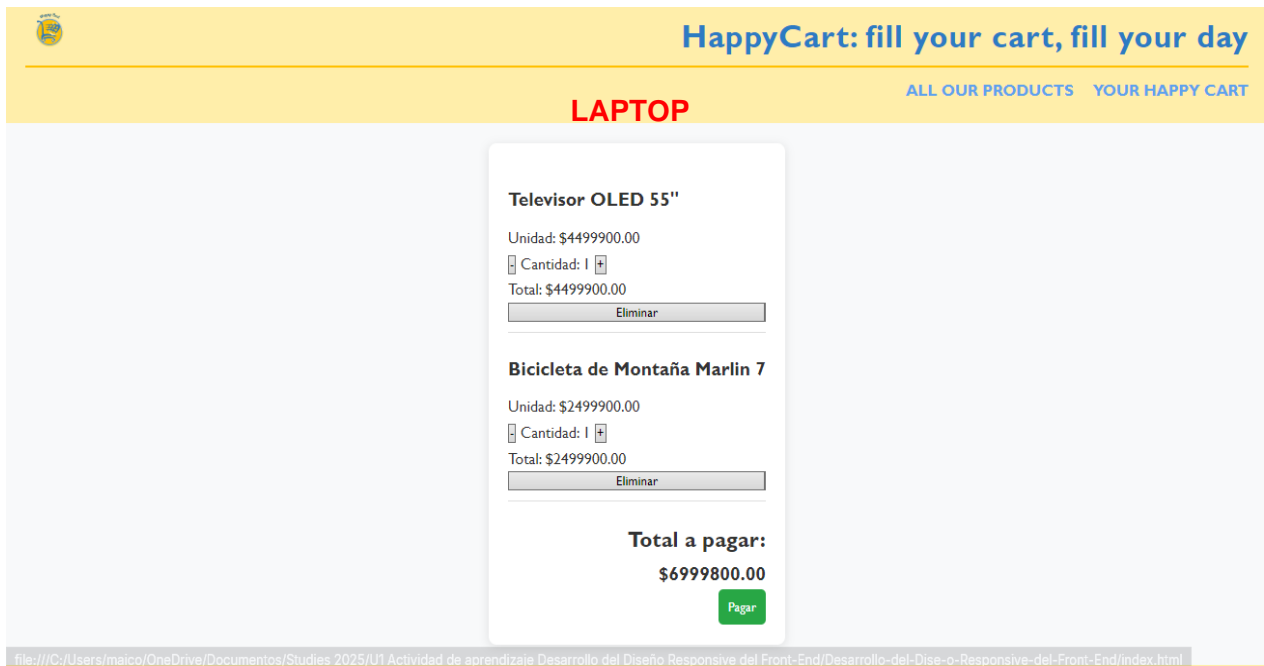
Nota. Una vez que abras la opción 'Inspeccionar', accederás automáticamente al modo de desarrollo web de Chrome, conocido como DevTools. Luego, busca el ícono situado al lado izquierdo de la opción 'Elements' en la barra de menú y selecciónalo.



Nota. Al seleccionar dicha opción, accederás al modo de Toggle Device Toolbar, donde podrás modificar las resoluciones y dimensiones de tu página web según los dispositivos

listados en el menú. Prueba con diferentes dispositivos móviles, como tabletas y teléfonos, para visualizar cómo se vería la página web en cada uno.





Conclusiones

Por lo anterior, al aplicar las diversas herramientas que ofrecen las tecnologías web, se puede lograr un diseño eficiente y fluido para la experiencia del usuario, priorizando siempre la información y presentándola de manera adecuada según las capacidades de cada portal. Por ello, el diseño responsive adquiere importancia en la creación de una página web.

El diseño responsive no solo adapta el contenido a las pantallas móviles, sino que también ofrece una navegación óptima para el usuario en estas plataformas. Al incluir botones con funciones adicionales, imágenes ilustrativas, contenido textual con una buena fuente y tamaño, y al mostrar la información con un estilo de layout propio, se puede marcar una gran diferencia entre el reconocimiento y el abandono de una página web.

Este trabajo demuestra la importancia de implementar esta tecnología y lo sencillo que puede ser utilizar sus herramientas para el diseño responsive.

Referencias

Use Chrome DevTools to troubleshoot issues with your ads not showing - Google AdSense

Help. (s. f.). <https://support.google.com/adsense/answer/10858959>

Seobility. (s. f.). *¿Que es una Media Query? - Seobility Wiki.*

https://www.seobility.net/es/wiki/Media_Query

Mariño, M. (2023, 27 julio). *CSS Grid: ventajas y primeros pasos*. El Blog de Dinahosting.

<https://dinahosting.com/blog/css-grid-ventajas-y-primeros-pasos/>

Lorca, J. B. (2024, 8 febrero). *¿Qué es flexbox y cómo utilizarlo?* SEOCOM - Agencia

SEO y Marketing Digital. <https://seocom.agency/blog/elementor-22671/>