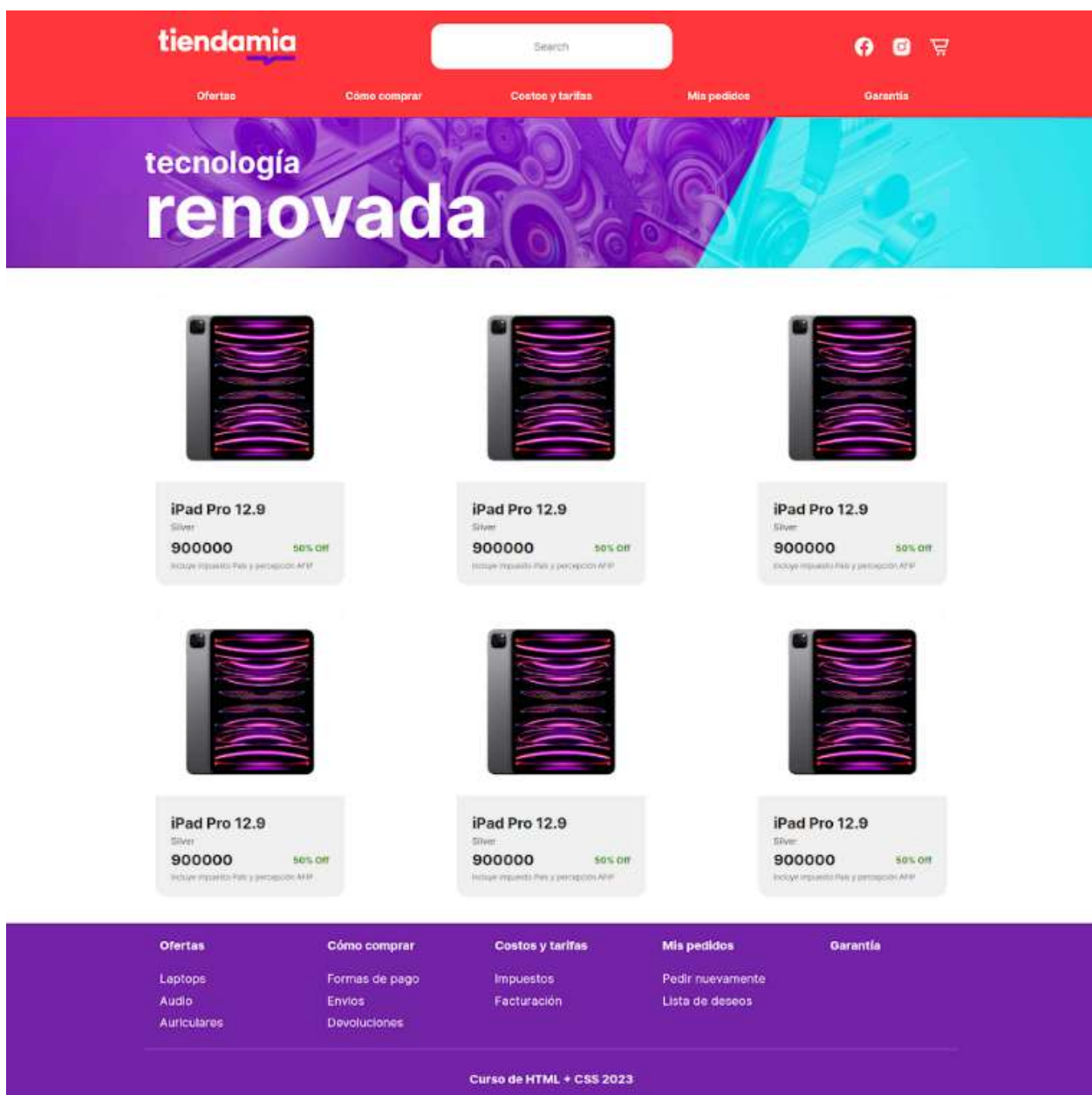


# 👣 Maquetado Básico de la Página

## 🚀 Introducción

En este paso, te enfocarás en establecer el maquetado básico de la página principal de tu tienda virtual.

Esta es la página que vas a crear (puedes descargar la maqueta en el link que está debajo)



Esta etapa es crucial para que comprendas cómo se estructurará y distribuirá el contenido en tu sitio. Utilizarás placeholders y bordes de colores para delinear

visualmente las áreas principales, lo que te permitirá tener una clara visualización del diseño general antes de proceder con detalles más específicos.

Esta aproximación te ayudará a identificar de manera temprana cualquier desafío de diseño y a asegurarte de que la estructura de la página sea intuitiva y accesible para tus usuarios, siempre respetando el prototipo provisto.

---

## **Material requerido**

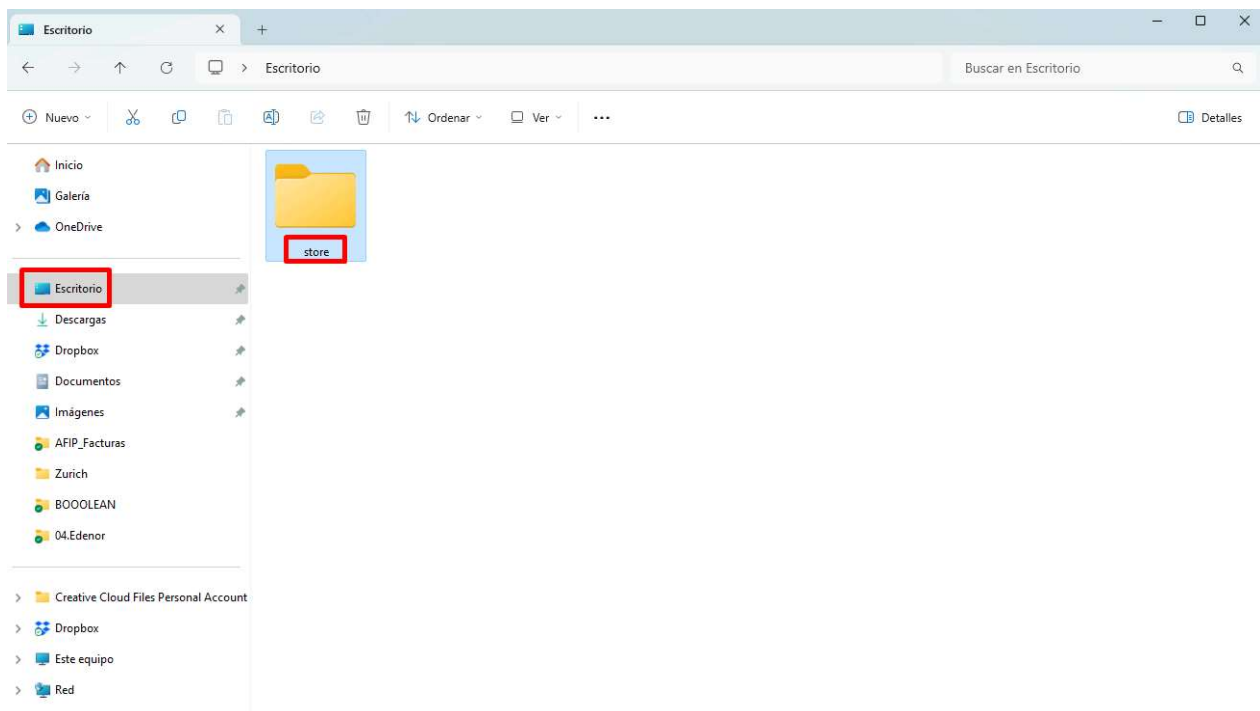
1. Prototipo de la tienda en formato JPG

 [Descarga prototipo vista principal de la tienda](#)

2. Carpeta starter con las imágenes del contenido inicial:

 [Descarga contenido](#)

*Te recomendamos buscar un espacio en tu computadora para crear la carpeta store. En esta carpeta trabajarás durante todo el curso.*



(en este ejemplo creamos la carpeta **store** en el escritorio)

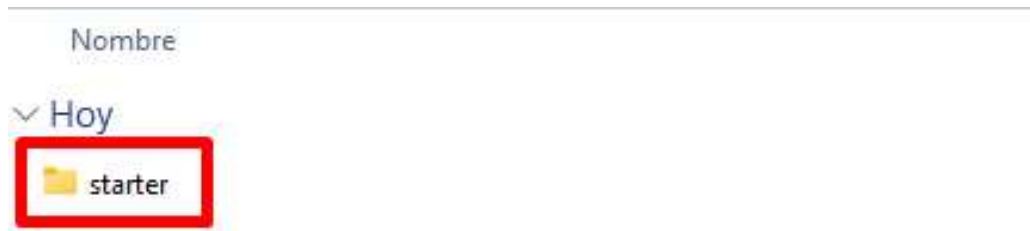
---

## **Actividades**

 **Actividad: Configurando el store**

Cuando descargues la carpeta verás la carpeta **starter**

En la carpeta **starter** verás el archivo **index.html**. Este archivo será la puerta de entrada a tu tienda virtual, donde vas a implementar los espacios reservados y bordes para el maquetado.



Dentro de tu archivo **index.html**, empieza por redactar la estructura básica de un documento HTML.

Esto abarca la declaración del tipo de documento (DOCTYPE), la etiqueta `<html>` con los atributos de lenguaje que correspondan, la sección `<head>`, donde incluirás información meta y el título de la página, y finalmente, la sección `<body>`, el espacio donde vas a colocar todo el contenido visible de tu página:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Tienda Virtual</title>
  </head>
  <body>
    Aquí desarrollaremos la estructura de nuestra tienda virtual.
  </body>
</html>
```

En esta etapa, no te preocupes por añadir estilos detallados o contenido específico. La idea es establecer una estructura básica que te permita visualizar la disposición general de la página según el prototipo provisto. Los detalles más finos y estilos específicos se añadirán en pasos posteriores.

*Una vez que tu archivo **index.html** esté configurado con esta estructura básica, habrás completado el primer paso crucial hacia el desarrollo de la página principal de tu tienda virtual.*

*Recuerda siempre guardar tus cambios a medida que avances en tu proyecto.*



## **Actividad: Creando la estructura principal**

Después de haber establecido el archivo `index.html` con la estructura básica del documento HTML, el siguiente paso es crear la estructura principal de la página. En este paso, definiremos las secciones clave que formarán el esqueleto de tu tienda virtual. Utilizaremos las etiquetas semánticas de HTML para organizar el contenido de manera lógica y accesible.

Podrás resolver esta actividad siguiendo las siguientes instrucciones:

1. Definir la Cabecera (Header): La etiqueta `<header>` de tu documento HTML será el área donde colocarás elementos como el logotipo de tu tienda, la barra de navegación o cualquier otro contenido que consideres pertinente para la cabecera de tu sitio. Piensa en esta sección como la primera impresión que tendrán los usuarios al visitar tu tienda.
2. Agregar Secciones Auxiliares (Sections): Utiliza la etiqueta `<section>` para crear distintas secciones auxiliares dentro de tu página. Estas pueden incluir banners promocionales, información adicional sobre productos, o cualquier otro elemento que complemente la experiencia de compra en tu tienda. En nuestro caso, la utilizaremos para crear un área publicitaria destacada con el nombre de la categoría de productos visibles.
3. Establecer la Sección Principal (Main): La etiqueta `<main>` es el corazón de tu página. Aquí se ubicará el contenido principal de tu tienda, como la exhibición de productos, descripciones, y cualquier otra información relevante para tus usuarios. Asegúrate de que esta sección sea clara y atractiva, ya que aquí es donde los usuarios pasarán la mayor parte del tiempo.
4. Incluir el Pie de Página (Footer): Finalmente, la etiqueta `<footer>` es donde incluirás información como enlaces a tus redes sociales, información de contacto, términos y condiciones, entre otros. El pie de página es un elemento importante para brindar confianza y opciones de contacto adicionales a tus usuarios.

Una vez hayas añadido estas etiquetas a tu documento HTML, tendrás una estructura principal sólida para tu tienda virtual. Asegúrate de guardar los cambios en tu archivo `index.html`. En los próximos pasos, nos enfocaremos en delimitar los límites de nuestros elementos HTML para diagramar la estructura de contenidos. El código de tu sitio debería verse así:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Tienda Virtual</title>
  </head>
  <body>
```

```
<header></header>
<section></section>
<main></main>
<footer></footer>
</body>
</html>
```

---



## **Actividad: Delimitando nuestro layout**

Ahora que hemos creado la estructura principal de nuestra página de tienda virtual, el siguiente paso es delimitar nuestro layout utilizando propiedades de estilo en CSS. En esta fase, vamos a definir los colores de los bordes y las dimensiones de las secciones principales utilizando CSS. Además, exploraremos conceptos como los valores de color en hexadecimal y la diferencia entre las propiedades `height` y `min-height`.

Podrás resolver esta actividad siguiendo las siguientes instrucciones:

1. Aplicando Estilos a las Secciones: Dentro del `<style>` en tu `<head>`, hemos definido los estilos para `header`, `section`, `main` y `footer`, asignando colores a los bordes para identificar visualmente cada sección y estableciendo sus respectivas alturas. Es importante recordar que estos estilos son temporales y sirven para visualizar la estructura del layout. En las fases de desarrollo futuro, estos estilos se ajustarán para alinearlos con el diseño final.
2. Diferencia entre `height` y `min-height`: Al definir las dimensiones de un elemento, podemos usar `height` para establecer una altura fija. Por ejemplo, en `<header>`, hemos definido una altura de `152px`. Esto significa que el encabezado siempre tendrá esta altura, sin importar el contenido que tenga dentro. Por otro lado, `min-height` se utiliza para establecer la altura mínima de un elemento, pero este puede expandirse si el contenido lo requiere. En el `<footer>`, hemos usado `min-height: 368px`, lo que significa que tendrá al menos esa altura, pero puede crecer si el contenido interno lo necesita. Es una propiedad útil para garantizar que sea responsivo y la adecuación del contenido.
3. Uso de Colores Hexadecimales: En CSS, los colores se pueden definir utilizando diferentes sistemas, y uno de los más comunes, además de los nombres de los colores principales, es el color hexadecimal. Un color hexadecimal es una forma de representar colores en formato digital, consistiendo en un símbolo `#` seguido de 6 caracteres (números y letras de la A a la F). Por ejemplo, en el caso del `<footer>`, utilizamos `#009145`, que es un código específico para un tono de verde.

4. Uso de height: auto: La propiedad height: auto; en el elemento `<main>` de tu HTML tiene un propósito importante en el diseño y la estructura de una página web. Al establecer `height: auto;`, estás permitiendo que la altura de este elemento se ajuste automáticamente para acomodar la cantidad de contenido que contiene.

El código de tu sitio debería verse así:

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Tienda Virtual</title>
  <style>
    header {
      border: 1px solid red;
      height: 152px;
    }
    section {
      border: 1px solid purple;
      height: 340px;
    }
    main {
      border: 1px solid blue;
      height: auto;
    }
    footer {
      border: 1px solid #009145; /* Hexadecimal color code*/
      min-height: 368px;
    }
  </style>
</head>
```

---



### **Actividad: Agregando un contenedor a la estructura**

Tras definir la estructura básica de nuestra tienda virtual y aplicar estilos iniciales a través de CSS, el siguiente paso es incorporar un elemento clave en el diseño web: el contenedor. Este paso se centrará en cómo agregar un contenedor para mejorar la disposición y alineación del contenido en nuestra página. Este contenedor servirá como un elemento de envoltura principal para cada sección, ayudando a mantener un diseño coherente y ordenado.

Podrás resolver esta actividad siguiendo las siguientes instrucciones:

1. Creando el Contenedor en HTML: Dentro de cada sección principal de nuestra página (`header`, `section`, `main`, `footer`), agrega un `<div>` con la clase `container`. Esta práctica es común en el diseño web para proporcionar un mejor control sobre cómo se muestra el contenido en diferentes dispositivos, manteniendo una consistencia visual en toda la página. Este contenedor actúa como un marco dentro del cual todo el contenido se alinearán y se organizará.

2. Estilizando el Contenedor en CSS: En la hoja de estilos integrada en nuestro archivo HTML, definiremos la clase `.container`. Esta clase aplicará un ancho fijo (`width: 1220px`), permitiendo que el contenido central de nuestra página no se extienda demasiado en pantallas grandes. Además, usaremos `margin: 0 auto` para centrar automáticamente el contenedor dentro de su elemento padre, y `border: 1px solid springgreen` para visualizar claramente sus límites durante la fase de desarrollo. Las propiedades de altura aseguran que el contenedor se adapte al contenido que contiene, manteniendo la consistencia del diseño.
3. Ajustando y Probando el Contenedor: Una vez que hayamos implementado los contenedores en nuestra página, el siguiente paso es ajustar y probar cómo interactúan con el contenido existente. Por ejemplo, en la sección `<main>`, ya hemos agregado algo de texto de muestra. Observaremos cómo este texto se alinea dentro del contenedor, lo que nos da una idea clara de cómo el contenido se visualizará en el contexto del diseño general de la página.
4. Refinamiento y Personalización: Finalmente, este paso no solo se trata de funcionalidad, sino también de estética y personalización. Aunque actualmente nuestro contenedor tiene un borde de color `springgreen` para fines de visualización, eventualmente querrás ajustar o eliminar este estilo para alinear con la identidad visual de tu tienda virtual. Este proceso de refinamiento es crucial para pasar de un prototipo funcional a un diseño final pulido y profesional.

*Asegúrate de revisar y ajustar tu contenedor en diferentes tamaños de pantalla para garantizar que tu diseño sea responsivo y accesible en una variedad de dispositivos.*

Una vez completado este paso, tendrás una base sólida y flexible para continuar desarrollando y estilizando el resto de tu tienda virtual, asegurando que todo el contenido esté bien organizado y presentado de manera efectiva. El código dentro de la etiqueta `<body>` debería verse del siguiente modo:

```
<header>
  <div class="container"></div>
</header>
<section>
  <div class="container"></div>
</section>
<main>
  <div class="container">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, voluptates.
    Quis fugit aut perspiciatis, tempora quos maxime doloremque similique
    voluptas. Repellendus cupiditate, autem ab a vero eveniet voluptatem animi
    excepturi!
  </div>
</main>
<footer>
```



```
<div class="container"></div>
</footer>
```

Por otro lado, el código dentro de tu etiqueta ``<style>`` debería reflejar la siguiente adición:

```
.container {
  width: 1220px;
  margin: 0 auto;
  border: 1px solid springgreen;
  height: inherit;
  min-height: inherit;
}
```

---



### **Actividad: Visualizando nuestra tienda por primera vez en el navegador**

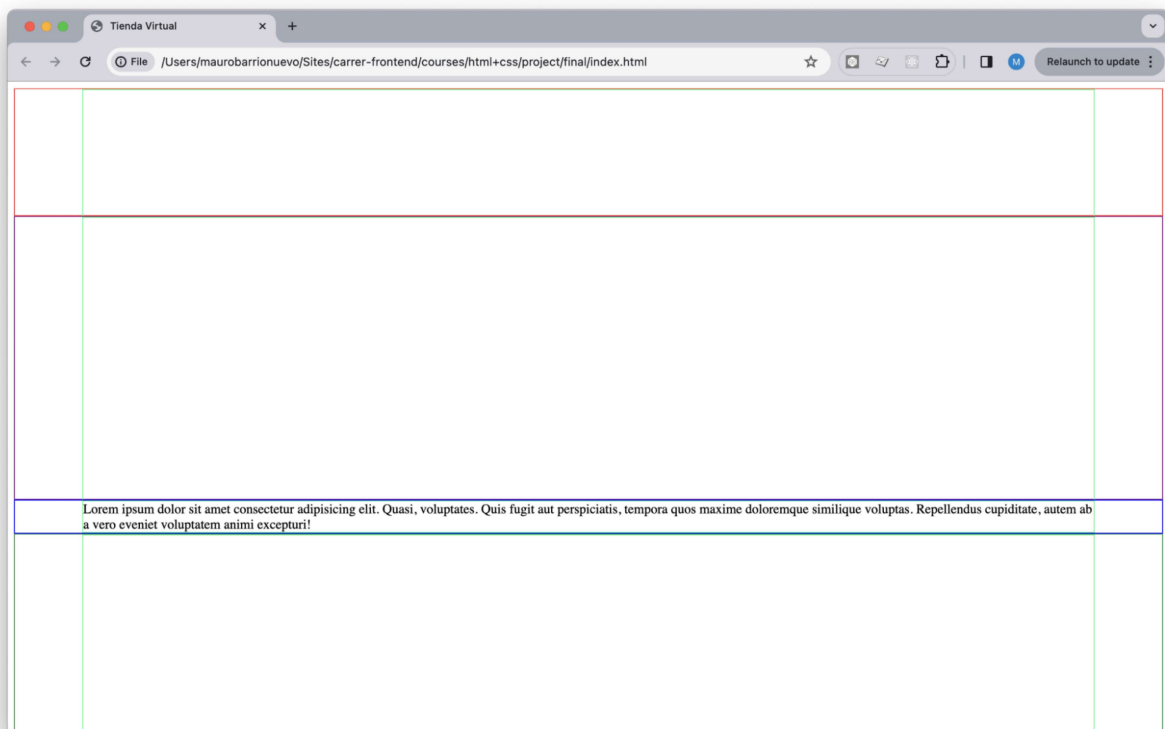
El objetivo de este paso es realizar una primera revisión visual de nuestra tienda virtual en un navegador web. Esto nos permitirá ver cómo se traducen nuestros códigos HTML y CSS en una página web real y nos ayudará a identificar áreas que requieran ajustes o mejoras.

Podrás resolver esta actividad siguiendo las siguientes instrucciones:

1. Abrir el Archivo en un Navegador: Para comenzar, necesitas abrir el archivo HTML de tu tienda virtual en un navegador web. Esto generalmente se puede hacer de dos maneras:
  - Haciendo doble clic en el archivo HTML si está almacenado localmente en tu computadora. Esto abrirá automáticamente el archivo en tu navegador predeterminado.
  - Utilizando un editor de código con capacidades de vista previa en vivo (como Visual Studio Code con la extensión Live Server), que reflejará en tiempo real los cambios que realices en tu código.
1. Inspección Visual: Una vez abierto el archivo en el navegador, realiza una inspección visual detallada de todos los elementos de la página. Presta especial atención a cómo se han aplicado los estilos de los bordes y las dimensiones que definiste anteriormente. Asegúrate de que cada sección (header, section, main, footer) se visualice claramente con los bordes y colores especificados.
2. Haz pruebas: Realiza pruebas agregando diferentes cantidades y tipos de contenidos a las distintas secciones. Puedes probar anidando elementos de otros tipos.



3. Revisar que sea Responsivo: Cambia el tamaño de la ventana de tu navegador para simular diferentes tamaños de pantalla, incluyendo tablets y smartphones. Observa cómo responde el layout a estos cambios. ¿Se ajustan las secciones de manera fluida? ¿El contenido se desborda en algún punto?
4. Identificar Posibles Problemas: Si notas algo que no se ve como esperabas (por ejemplo, secciones que no se alinean correctamente, colores que no coinciden, o contenido desbordado), toma nota de estos problemas. Estos serán puntos clave para trabajar en los siguientes pasos del desarrollo.
5. Captura de Pantalla (Opcional): Podría ser útil tomar capturas de pantalla de la vista actual para discutir con el equipo o para tener una referencia visual de cómo se veía la página en esta etapa del desarrollo. Debería verse similar a esta:



Al final de este paso, deberías tener una comprensión clara de cómo se está materializando tu diseño web en un entorno real del navegador. Esta visualización inicial es fundamental para orientar las próximas fases de desarrollo, ajustando lo necesario para asegurar que el diseño final se alinee con tu visión original y las expectativas de los usuarios.

---

## **Actividades finalizadas**

- Creando el archivo HTML

- Creando la estructura principal
  - Delimitando nuestro layout
  - Agregando un contenedor a la estructura
  - Visualizando nuestra tienda por primera vez en el navegador
- 



## **Información complementaria**

- [Introducción a HTML](#)
- [Primeros pasos con HTML](#)
- [Estructura y documentación](#)
- [Primeros pasos en CSS](#)
- [Cómo se estructura CSS](#)
- [Selectores CSS](#)