

Capítulo 3: Fundamentos de Hojas de Estilo en Cascada (CSS)

MSc. Juan Antonio Castro Silva

May 2, 2020

Versión: 0.1 (20200315_0926)

1 Introducción

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje utilizado para describir la presentación de documentos HTML. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios.

Como HTML, CSS no es realmente un lenguaje de programación. Es un lenguaje de hojas de estilo, es decir, permite aplicar estilos de manera selectiva a elementos en documentos HTML.

CSS es uno de los lenguajes base de la Open Web y posee una especificación estandarizada por parte del W3C. Desarrollado en niveles, CSS1 es ahora obsoleto, CSS2.1 es una recomendación y CSS3, ahora dividido en módulos más pequeños, está progresando en camino al estándar.

En este capítulo, se discutirán las características básicas de CSS y como utilizarlo en la práctica. El código fuente de este capítulo se encuentra en el repositorio de github del curso https://github.com/juancasi/programacion_web.

2 Proyecto

Uno de los proyectos de este curso es una aplicación web de comercio electrónico (tienda virtual). Los conceptos básicos de CSS serán aplicados en el desarrollo del proyecto planteado (aprender haciendo-learn by doing). El proyecto principal de este capítulo es la versión 1, primera aproximación, de una página web que muestre los productos de la tienda, utilizando únicamente HTML y CSS.

2.1 Diseño

Se requiere desarrollar una página web que corresponda al diseño especificado en la figura 1.

2.2 Implementación

El código HTML del listado 1 corresponde al contenido y la estructura especificado en el diseño de la figura 1.

Listing 1: /src/estructura/index.html (Estructura)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Tienda Virtual - V1</title>
5 <link rel="stylesheet" type="text/css" href="css/estilo_01.css">
6 </head>
7 <body>
8
9 <header>
10 <h1>Nombre de la Empresa</h1>
11 <nav>
12 <ul class="menu_hor">
13 <li><a href="index.html">Inicio</a></li>
14 <li><a href="pedidos.html">Pedidos</a></li>
15 <li><a href="contactenos.html">Contactenos</a></li>
16 </ul>
17 </nav>
18 </header>
19
```



Figure 1: Diseño

```

20 <div>
21 <section>
22 <article>
23 
24 <h3>Nombre del producto 1</h3>
25 <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
    arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
    pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
26 </article>
27
28 <article>
29 
30 <h3>Nombre del producto 2</h3>
31 <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
    arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
    pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
32 </article>
33
34 <article>
35 
36 <h3>Nombre del producto 3</h3>
37 <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
    arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
    pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
38 </article>
39
40 </section>
41
42 <aside>
43 
44 <br/>

```

```
45 
46 <br/>
47 
48 <br/>
49 </aside>
50 </div>
51
52 <footer>
53 <h2>Nombre de la Empresa</h2>
54 <h3>Dirección: Carrera 76 # 23 - 98</h3>
55 <h3>Teléfonos: 8776655 315-098-7654</h3>
56 </footer>
57
58 </body>
59 </html>
```

2.3 Prueba

Para testear la implementación de los elementos de estructura, abra en un browser el archivo `chapter_03/src/estructura/index.html`.

En la figura 2 se observa que la apariencia del contenido resultante no corresponde con el diseño planteado. Para lograr la apariencia bosquejada en el diseño se debe hacer uso de las hojas de estilo en cascada (CSS).

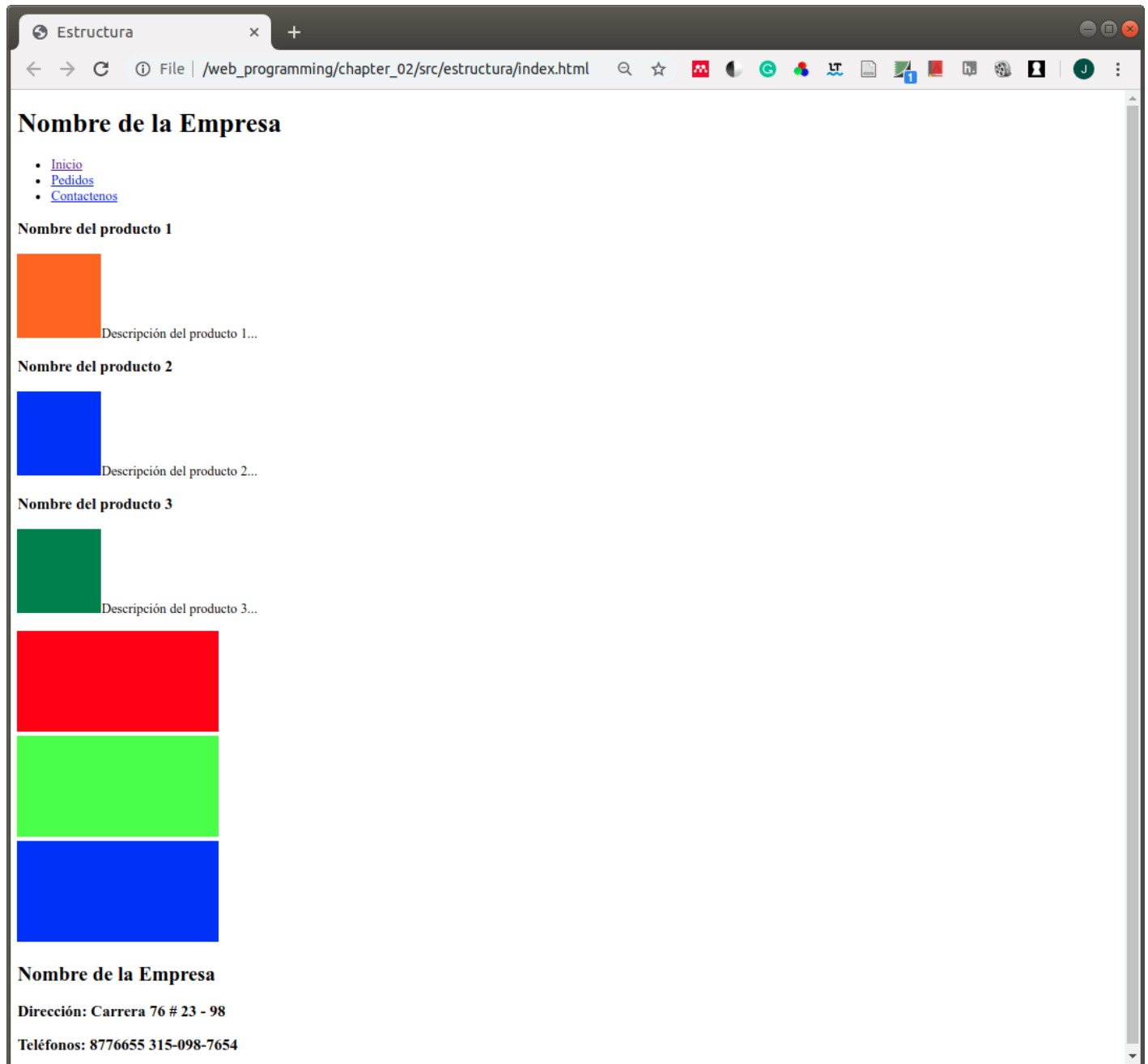


Figure 2: Estructura

3 Sintaxis CSS

Un CSS consiste de un conjunto de reglas (rule-set) de un selector y un bloque de declaración:

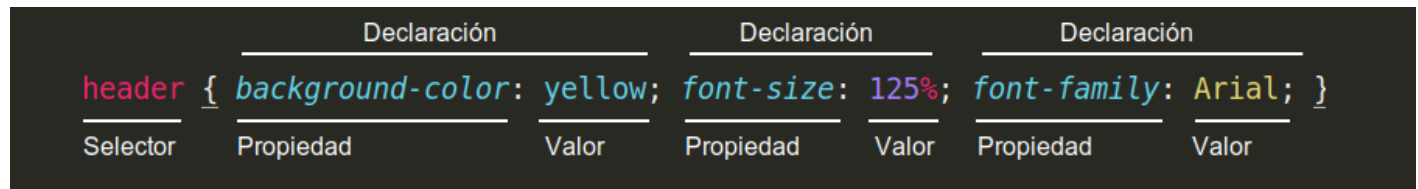


Figure 3: Sintaxis CSS

El selector apunta al elemento HTML que se desea estilizar (aplicar un estilo). El bloque de declaración incluye una o más declaraciones separadas por punto y coma(;). Cada declaración incluye un nombre de propiedad CSS y un valor, separado por dos puntos (:). Múltiples declaraciones de CSS son separadas con puntos y comas, y los bloques de declaración son rodeados por llaves ({}).

Explicación del ejemplo:

- header es un selector CSS (apunta al elemento HTML que se desea estilizar: header).
- background-color es una propiedad, y yellow es el valor de la propiedad.
- font-size es una propiedad, y 125% es el valor de la propiedad.
- font-family es una propiedad, y Arial es el valor de la propiedad.

4 Tipos de CSS

Existen tres formas de añadir código CSS a una página web.

4.1 CSS Inline (NO recomendado)

El tipo de CSS Inline aplica el estilo directamente a cada elemento HTML.

Listing 2: /src/intro/css.inline_01.html (CSS Inline)

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Inline</title>
5 <meta charset="utf-8"/>
6 </head>
7 <body>
8 <header style="background-color: yellow; font-size: 125%; font-family: Arial;">
9   <h1>Nombre de la Empresa</h1>
10 </header>
11 </body>
12 </html>
  
```

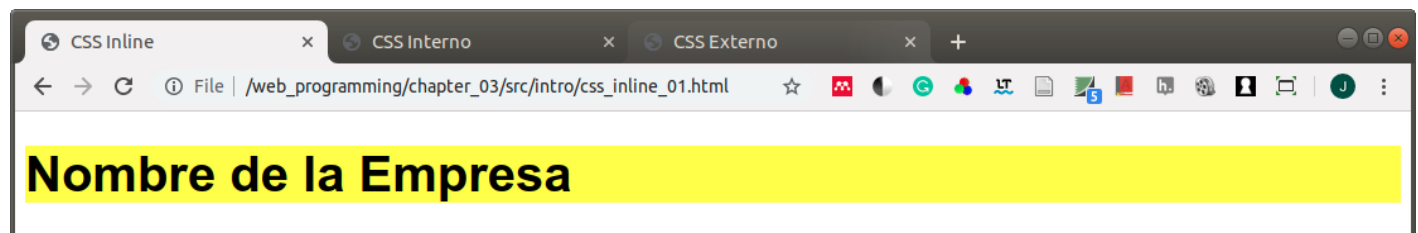


Figure 4: CSS Inline

4.2 CSS Interno (NO recomendado)

El tipo de CSS Interno define las reglas de estilo que se aplican en una página web.

Listing 3: /src/intro/css_interno_01.html (CSS Interno)

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Interno</title>
5 <meta charset="utf-8"/>
6 <style type="text/css">
7     header {
8         background-color: yellow;
9         font-size: 125%;
10        font-family: Arial;
11    }
12 </style>
13 </head>
14 <body>
15 <header>
16     <h1>Nombre de la Empresa</h1>
17 </header>
18 </body>
19 </html>

```

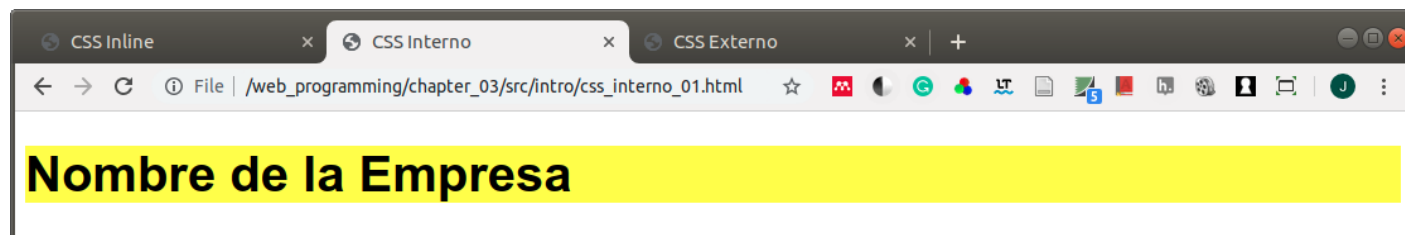


Figure 5: CSS Interno

4.3 CSS Externo (Recomendado-Best Practice)

El tipo de CSS Externo define el conjunto de reglas de estilo que se aplican a un sitio web. Es el modelo recomendado por multiples razones:

1. Al cambiar una regla de estilo se afecta a todas las páginas del sitio. No se tiene que cambiar en cada elemento o pagina como en los dos modelos anteriores (CSS Inline y CSS Interno).
2. Permite el reuso.
3. Mantenimiento más facil.
4. Posibilita el trabajo en equipo (expertos en diseño gráfico).
5. Permite separar las diferentes capas (contenido-estructura=HTML, apariencia=CSS y comportamiento=JavaScript).

El documento HTML define el contenido y la estructura de la página web. Para vincular un hoja de estilo en cascada (CSS) a un documento HTML se emplea la etiqueta <link>, el atributo href="..." especifica el nombre del archivo CSS.

Listing 4: /src/intro/css_externo_01.html (CSS Externo)

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Externo</title>
5 <meta charset="utf-8"/>
6 <link rel="stylesheet" type="text/css" href="css/estilo_01.css">
7 </head>
8 <body>
9 <header>
10    <h1>Nombre de la Empresa</h1>

```

```

11 </header>
12 </body>
13 </html>

```

El documento CSS define las reglas de estilo que se desean aplicar (apariciencia).

Listing 5: /src/intro/css/estilo_01.css (CSS Externo)

```

1 header {
2   background-color: yellow;
3   font-size: 125%;
4   font-family: Arial;
5 }

```

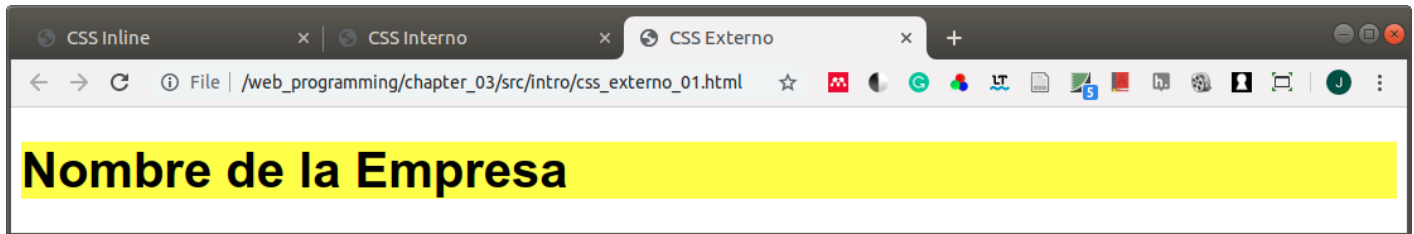


Figure 6: CSS Interno

5 Selectores

Los selectores CSS se usan para "encontrar" (o seleccionar) los elementos HTML a los que se les desea dar estilo (aplicar un estilo).

Los selectores se pueden agrupar en cinco (5) categorías:

1. Simple selectors (select elements based on name, id, class)
2. Combinator selectors (select elements based on a specific relationship between them)
3. Pseudo-class selectors (select elements based on a certain state)
4. Pseudo-elements selectors (select and style a part of an element)
5. Attribute selectors (select elements based on an attribute or attribute value)

En este capítulo se explicarán los selectores más básicos.

5.1 HTML de Ejemplo

Los ejemplos de los selectores darán estilo al código HTML definido en el listado 6

Listing 6: HTML ejemplo selectores

```

1 <h1>Selectores</h1>
2
3 <h2>Sección 1</h2>
4 <article>
5 <p class="sangria">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula
   eget dolor. Aenean massa.</p>
6 <p>Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.</p>
7 <p>Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis
   enim.</p>
8 </article>
9
10 <h2>Sección 2</h2>
11 <div>
12 <p class="sangria">Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim
   justo, rhoncus ut, imperdiet a, venenatis vitae, justo.</p>
13 <p>Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum
   semper nisi. Aenean vulputate eleifend tellus.</p>

```

```
14 <p>Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus  
    in, viverra quis, feugiat a, tellus.</p>  
15 <p id="nota">Este es un ejemplo de id</p>  
16 </div>
```

5.2 Selector tipo elemento

El selector tipo elemento selecciona los elementos HTML basado en el nombre del elemento.

```
1 p {  
2   font-family: Arial;  
3   color: blue;  
4 }
```

En este ejemplo las reglas de estilo indican que todos los elementos `<p>`, todos los párrafos, tendrán una fuente de tipo Arial (`font-family: Arial;`) y un color de letra azul (`color: blue;`), como se muestra en la figura 7.

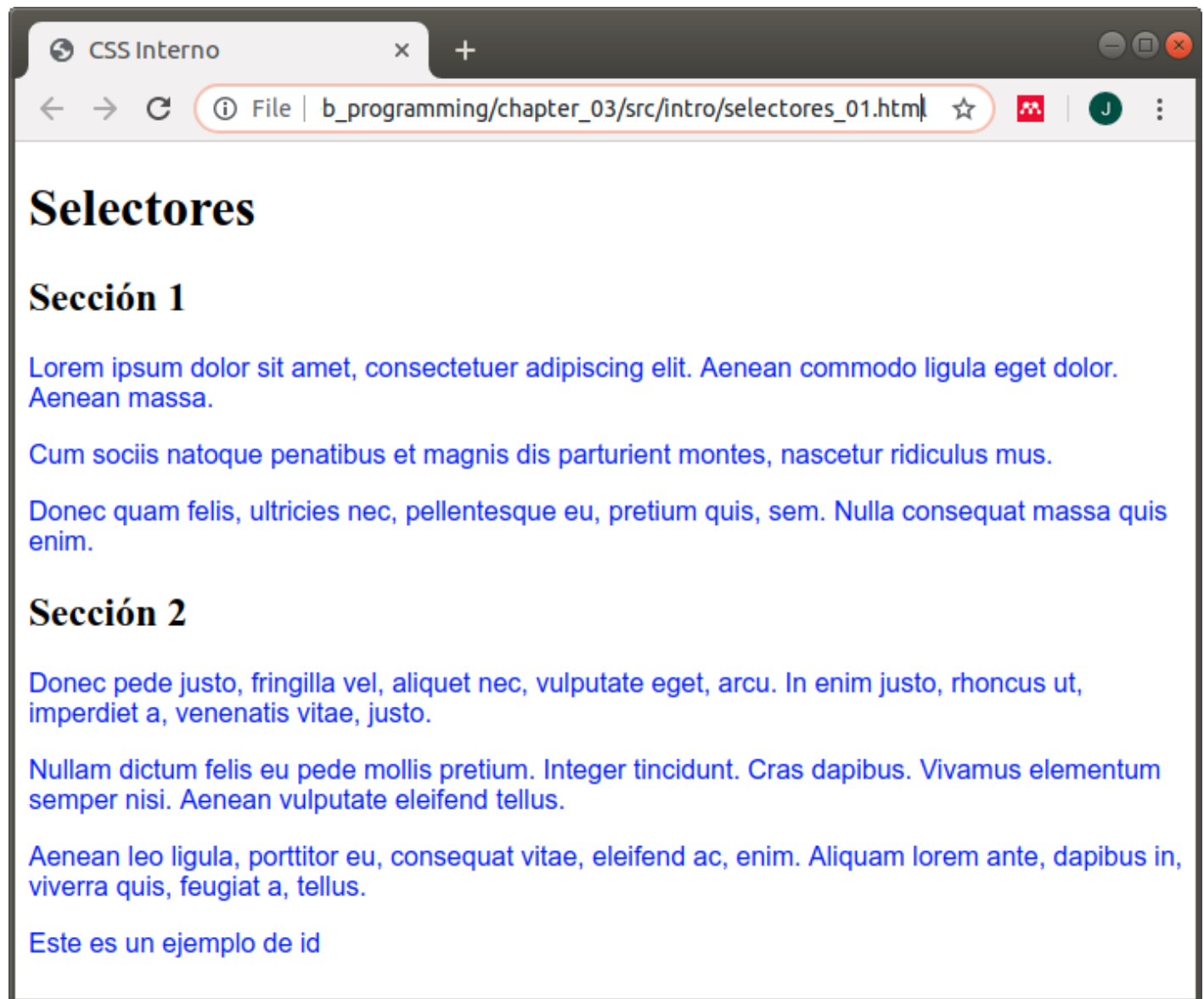


Figure 7: Selector tipo elemento

5.3 Selector tipo clase

El selector tipo class selecciona elementos HTML con un atributo class específico. Para seleccionar elementos con una clase específica, escriba un caracter punto (.), seguido por el nombre de la clase, por ejemplo (.sangria {...}).

```
1 .sangria {  
2   padding-left: 50px;  
3 }
```

En este ejemplo todos los elementos HTML con la clase="sangria" tendrán un espacio a la izquierda de 50 pixels (padding-left: 50px;), como se muestra en la figura 8.

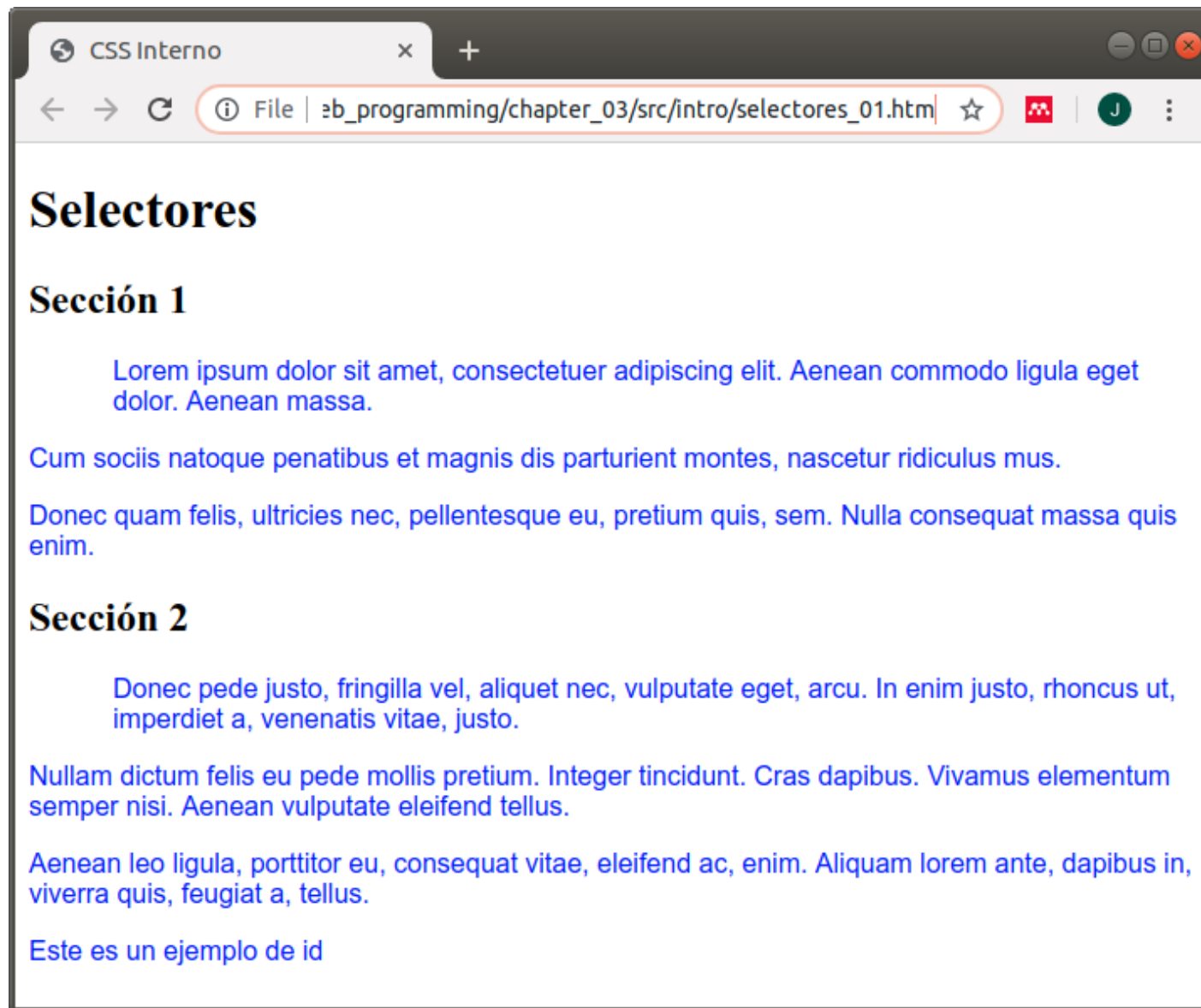


Figure 8: Selector tipo clase

5.4 Selector tipo id

El selector tipo id utiliza el atributo id de un elemento HTML para seleccionar un elemento específico. El atributo id de un elemento es unico dentro de una página, de modo que el selector id es utilizado para seleccionar un elemento único. Para seleccionar un elemento con un id específico, escriba un caracter numeral (#), seguido por el id del elemento, por ejemplo (#nota {...}).

```
1 #nota {
```

```
2 color: red;
3 font-weight: bold;
4 }
```

En este ejemplo la regla CSS será aplicada al elemento con el id="nota", el elemento tendrá un color de fuente rojo (color: red;) y el texto estará en negrilla (font-weight: bold;), como se muestra en la figura 9.

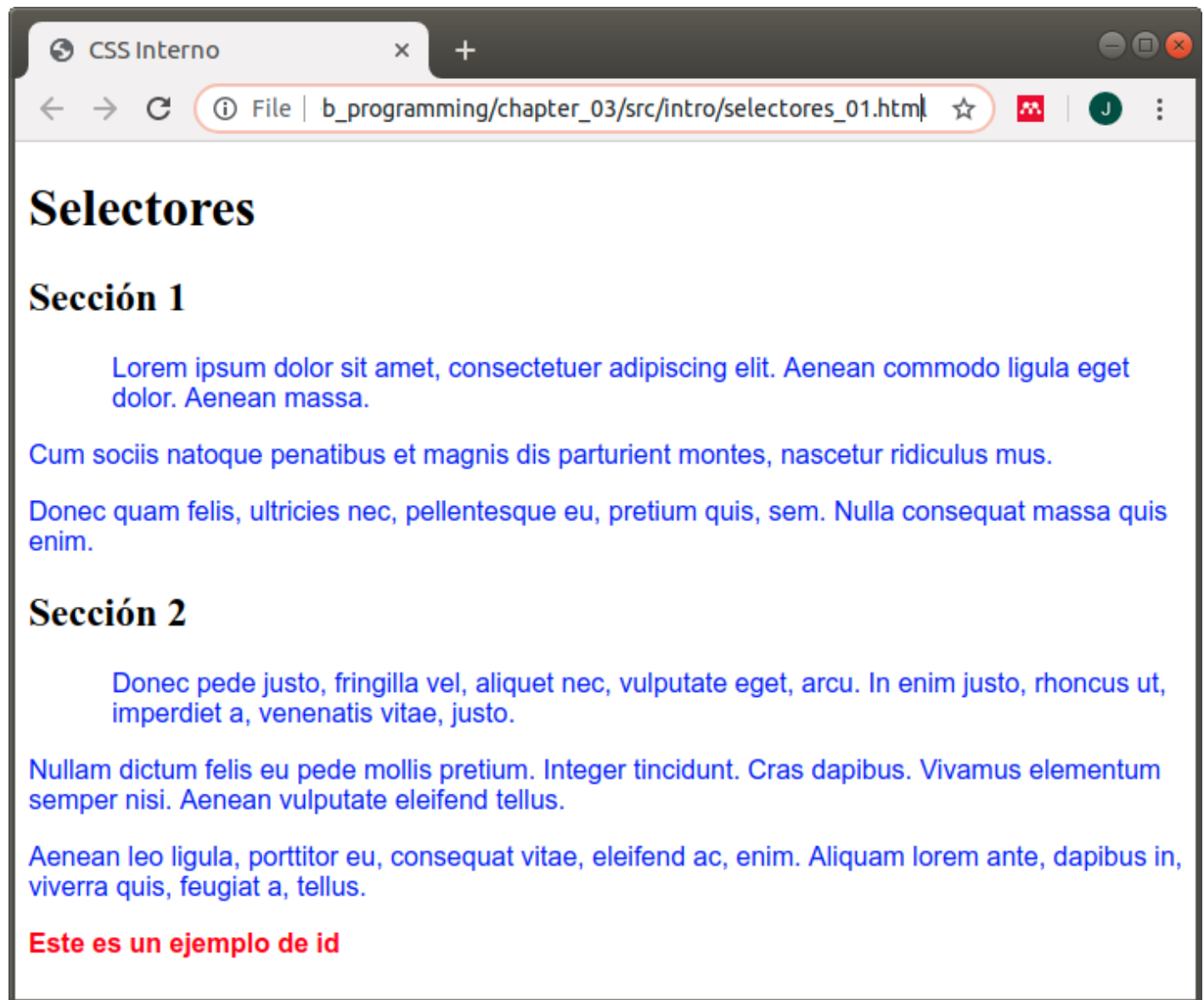


Figure 9: Selector tipo id

5.5 Agrupar selectores

El selector de agrupamiento selecciona todos los elementos HTML con las mismas definiciones de estilo. En el siguiente código CSS los elementos h1 y h2 tienen las mismas definiciones de estilo, por lo tanto será mejor agrupar los selectores, para minimizar el código. Para agrupar selectores, separe cada selector con una coma.

```
1 h1 {
2   background-color: yellow;
3   border-left: solid blue 10px;
4   padding-left: 20px;
5 }
6 h2 {
7   background-color: yellow;
8   border-left: solid blue 10px;
```

```
9 padding-left: 20px;  
10 }
```

En este ejemplo se han agrupado los elementos del código de arriba (h1 y h2).

```
1 h1, h2 {  
2 background-color: yellow;  
3 border-left: solid blue 10px;  
4 padding-left: 20px;  
5 }
```

A los dos elementos (h1 y h2) se les aplicará un color de fondo amarillo (background-color: yellow;), tendrán a la izquierda un borde solido azul de 10 pixels (border-left: solid blue 10px;) y el texto será espaciado 20 pixels a la izquierda (padding-left: 20px;), como se muestra en la figura 10.



Figure 10: Agrupar selectores

5.6 Modelo de caja (Box Model)

Todos los elementos HTML se pueden considerar como cajas (cuadros). En CSS, el termino "modelo de caja" (box model) se utiliza cuando se esta hablando de diseño y maquetación.

El modelo de caja en CSS es esencialmente un cuadro que envuelve todos los elementos HTML. Consiste en: márgenes, bordes, relleno (espacia) y el contenido real. La figura 11 ilustra el modelo de caja:

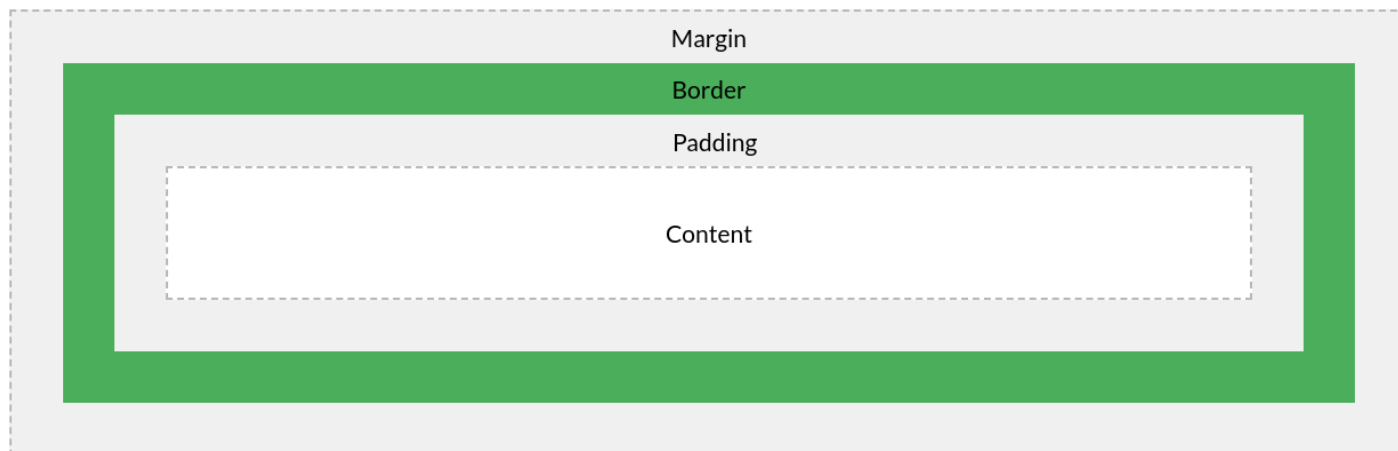


Figure 11: Modelo de caja (Box Model)

Explicación de las diferentes partes:

- Contenido (Content): El contenido del cuadro, donde aparecen el texto y las imágenes.
- Relleno (Padding): Borra un área alrededor del contenido. El relleno (espaciado) es transparente.
- Borde (Border): Un borde que rodea el relleno y el contenido.
- Margen (Margin): Borra un área fuera del borde. El margen es transparente.

5.6.1 Ejemplo

Para aplicar los conceptos del modelo de caja, en este ejemplo se van a definir unas reglas de estilo para aplicarlas al elemento header.

Listing 7: /src/intro/box_model_01.html (HTML del elemento header)

```

1 <header>
2   <h1>Nombre de la Empresa</h1>
3   <nav>
4     <ul>
5       <li><a href="index.html">Inicio</a></li>
6       <li><a href="pedidos.html">Pedidos</a></li>
7       <li><a href="contactenos.html">Contactenos</a></li>
8     </ul>
9   </nav>
10 </header>

```

Las reglas de estilo definen que el elemento header tendrá un color de fondo amarillo (background-color: yellow;) y un tipo de fuente Arial (font-family: Arial;). Los elementos <h1> que esten dentro del <header> (header h1 ({...}) tendrán un margen de 0 pixel (margin: 0px;), un espacio de relleno de 10 pixels (padding: 10px;) y un tamaño de fuente de 250% (font-size: 250%;). Los elementos que esten dentro del elemento <nav> tendrán un margen de 0 pixels (margin: 0px;) y un color de fondo verde (background-color: green;).

Listing 8: /src/intro/css/box_model_01.css (CSS del elemento header)

```

1 header {
2   background-color: yellow;
3   font-family: Arial;
4 }
5 header h1 {
6   margin: 0px;
7   padding: 10px;
8   font-size: 250%;
9 }
10 nav ul {
11   margin: 0px;
12   background-color: green;
13 }

```

La figura 12 muestra como el margen del elemento body, asignado por defecto por el navegador, impide que el elemento header ocupe todo el ancho disponible (100%).

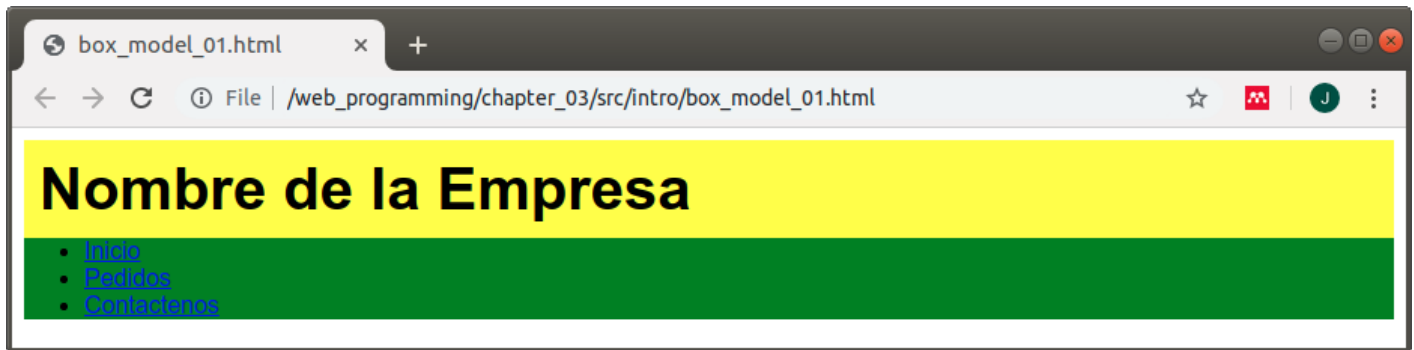


Figure 12: Modelo de caja

Para quitar el margen por defecto del elemento body, se adiciona una regla de estilo (body {margin: 0px;}).

Listing 9: /src/intro/css/box_model.01.css (CSS del elemento header)

```
1 body {
2   margin: 0px;
3 }
```

La figura 13 muestra el elemento header ocupando el 100% del ancho de la página.



Figure 13: Modelo de caja (body {margin: 0px;})

6 Barra de navegación (nav)

El código HTML correspondiente a la barra de navegación esta contenido en el elemento <nav>. El menú se define con la ayuda del elemento .

```
1 <nav>
2 <ul class="menu_hor">
3 <li><a href="index.html">Inicio</a></li>
4 <li><a href="pedidos.html">Pedidos</a></li>
5 <li><a href="contactenos.html">Contactenos</a></li>
6 </ul>
7 </nav>
```

Para darle estilo a la barra de navegación se aplicarán las siguientes reglas. Para mayor facilidad y posibilitar el reuso de las reglas de estilo que definen el menú horizontal se ha creado la clase menu horizontal (class="menu_hor") que corresponde al elemento que esta dentro del elemento <nav>.

```
1 /* --- barra de navegación ---*/
2 .menu_hor {
3   list-style-type: none;
4   margin: 0px;
5   padding: 0px;
```

```

6  overflow: hidden;
7  background-color: green;
8  }

```

Al elemento `(.menu_hor {...})` se le suprimen las viñetas que se muestran por defecto (`list-style-type: none;`), se elimina el margen (`margin: 0;`), se borra el espacio de relleno (`padding: 0;`), se ocultan las barras de desplazamiento (`overflow: hidden;`) y se le asigna un color de fondo verde (`background-color: green;`).

```

1  .menu_hor li {
2    float: left;
3    border-right: 1px solid yellow;
4  }

```

Los elementos de lista ` (.menu_hor li {...})` se hacen flotar a la izquierda, para que se muestren horizontalmente uno al lado del otro (`float: left;`) y se define un borde solido de un pixel de color amarillo (`border-right: 1px solid yellow;`).

```

1  .menu_hor li a {
2    display: block;
3    color: white;
4    text-align: center;
5    padding: 12px 14px;
6    text-decoration: none;
7  }

```

Los enlaces correspondientes a los elementos `<a>` que estan dentro de cada elemento de lista ` (.menu_hor li a {...})` se despliegan como un bloque (`display: block;`), para que parezcan un boton rectangular, se les asigna un color de texto blanco (`color: white;`), se alinea el texto para que aparezca centrado (`text-align: center;`), se asigna un espacio de relleno (`padding: 12px 14px;`) y se suprime el subrayado por defecto del enlace (`text-decoration: none;`).

```

1  .menu_hor li:last-child {
2    border-right: none;
3  }

```

Al último elemento de lista ` (.menu_hor li:last-child {...})` se le suprime el borde derecho (`border-right: none;`). En la figura 14 se muestra la barra de navegación horizontal.

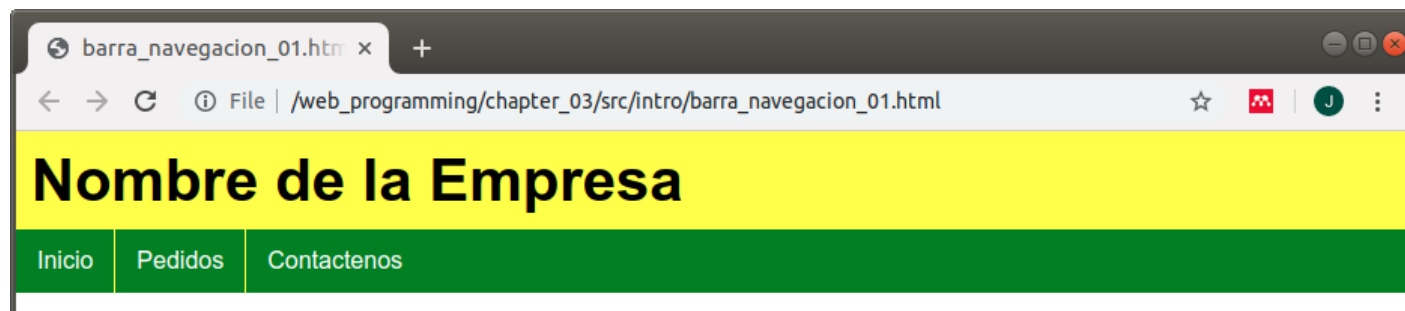


Figure 14: Barra de navegación

7 Posicionamiento

Para posicionar los elementos HTML se pueden utilizar diversas propiedades de CSS (float, left, right, top, bottom, entre otras). En este ejemplo vamos utilizar float para darle estilo al código HTML que contiene una imagen y texto sin formato.

```
1 <article>
2 
3 <h3>Nombre del producto 1</h3>
4 <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
   arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
   pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
5 </article>
```

La figura 15 muestra la representación del código sin aplicar CSS.



Figure 15: HTML original sin CSS

Para hacer que la imagen flote a la izquierda, se definen unas reglas de estilo como se muestra en el listado 10.

Listing 10: /src/intro/css/float_01.css

```
1 article img {
2   float: left;
3   width: 100px;
4   height: 100px;
5   margin-right: 10px;
6 }
```

Las reglas de estilo definen que la imagen flotará a la izquierda (float: left;), tendrá un ancho y un alto de 100 píxeles (width: 100px; height: 100px;) y que se aplicará un margen a la derecha de 10 píxeles (margin-right: 10px;), como se muestra en la figura 16. Es importante resaltar la línea 1 (article img {...}), define que las reglas de estilo se aplicarán a todos los elementos que estén dentro del elemento <article>.

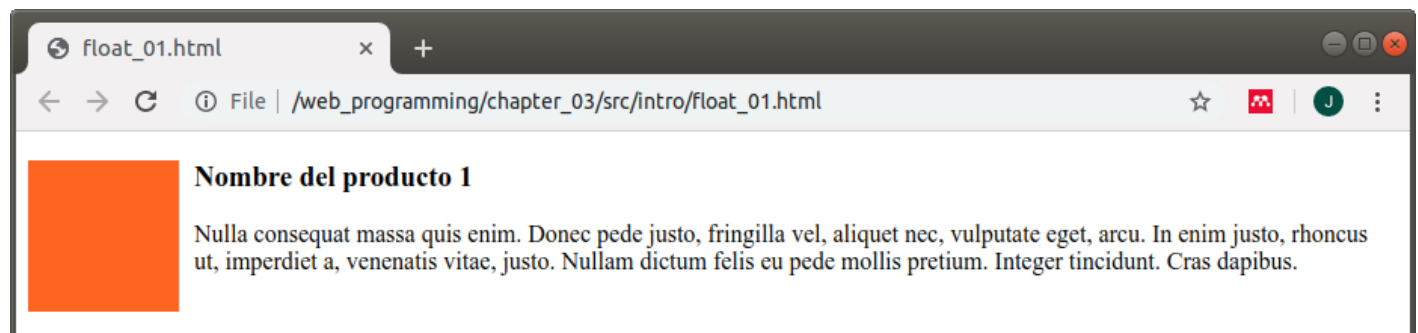


Figure 16: HTML con CSS

8 Maquetación con CSS (Layout)

El listado 11 corresponde al código HTML que define la estructura y contenido de los componentes que muestran los productos (<section>) y los banners (<aside>).

Listing 11: /src/intro/float_02.html (Componentes productos y banners)

```

1 <div>
2 <section>
3   <article>
4     
5     <h3>Nombre del producto 1</h3>
6     <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
       arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
       pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
7   </article>
8
9   <article>
10    
11    <h3>Nombre del producto 2</h3>
12    <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
       arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
       pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
13  </article>
14
15  <article>
16    
17    <h3>Nombre del producto 3</h3>
18    <p>Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
       arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu
       pede mollis pretium. Integer tincidunt. Cras dapibus.</p>
19  </article>
20 </section>
21 <aside>
22   
23   <br/>
24   
25   <br/>
26   
27   <br/>
28 </aside>
29 </div>

```

El listado 13 corresponde al código CSS que permite dar el estilo deseado a los elementos <section> y <aside>.

```

1 section {
2   width: 73%;
3   float:left;
4 }
5 aside {
6   width: 23%;
7   float:left;
8 }
9 article {
10  clear: both;
11  border-bottom: dashed gray 1px;
12 }
13 article img {
14  float: left;
15  width: 100px;
16  height: 100px;
17  margin-right: 10px;
18 }
19 aside img {
20  margin-left:10px;
21 }

```

Los elementos (<section>) y (<aside>) se hacen flotar a la izquierda, para que queden uno al lado del otro (float:left;) y se definen los anchos (width: 73%; y width: 23%; respectivamente).

```

1 section {
2   width: 73%;
3   float:left;

```



```

4 }
5 aside {
6     width: 23%;
7     float: left;
8 }

```

El elemento `article` se limpia tanto en el sentido vertical como horizontal (`clear: both;`) y se le aplica un borde en la parte inferior (`border-bottom: dashed gray 1px;`).

```

17 article {
18     clear: both;
19     border-bottom: dashed gray 1px;
20 }

```

Los elementos `` que estan dentro del elemento `<article>` (`article img {...}`) se hacen flotar a la izquierda (`float: left;`), se les asigna un ancho y alto de 100 pixels (`width: 100px; height: 100px;`) y se les define un margen derecho de 10 pixels (`margin-right: 10px;`) para separar el texto de la imagen.

Listing 12: `/src/intro/css/float_02.css` (CSS para maquetación)

```

1 article img {
2     float: left;
3     width: 100px;
4     height: 100px;
5     margin-right: 10px;
6 }

```

Los elementos `` que estan dentro del elemento `<aside>` (`aside img {...}`) se definen con un margen izquierdo de 10 pixels (`margin-left: 10px;`) para separarlos del elemento `<section>`.

Listing 13: `/src/intro/css/float_02.css` (CSS para maquetación)

```

1 aside img {
2     margin-left: 10px;
3 }

```

La figura 17 muestra el diseño - maquetación (Layout) deseado.

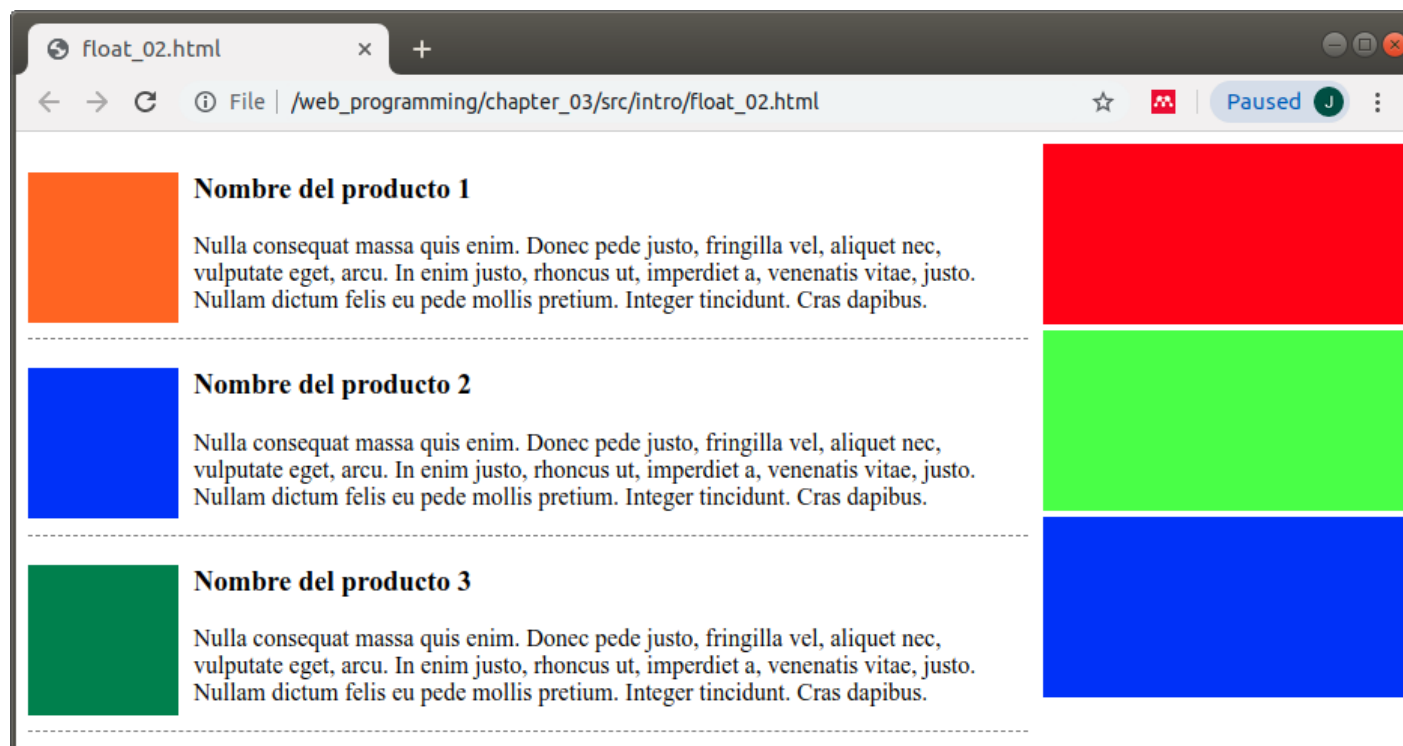


Figure 17: Maquetación con CSS (Layout)

9 Pie de página (Footer)

El elemento `<footer>` contiene elementos de encabezado `<h2>` y `<h3>` que permiten asignar el contenido del pie de página (Footer).

Listing 14: `/src/intro/footer_01.html` (HTML del footer)

```
1 <footer>
2 <h2>Nombre de la Empresa</h2>
3 <h3>Dirección: Carrera 76 # 23 - 98</h3>
4 <h3>Teléfonos: 8776655 315-098-7654</h3>
5 </footer>
```

Al elemento `<footer>` se le aplican reglas de estilo que hacen que el elemento sea movido hacia abajo para limpiar tanto elementos flotantes de la izquierda como de la derecha (`clear: both;`), se le define un color de fondo amarillo (`background-color: yellow;`), un margen de 0 pixels (`margin: 0px;`), un espacio de relleno de 5 pixels (`padding: 5px;`), un borde negro solido de 2 pixels en la parte superior (`border-top: 2px solid black;`) y un tipo de fuente Arial (`font-family: Arial;`).

Listing 15: `/src/intro/css/footer_01.css` (CSS del footer)

```
1 footer {
2   clear: both;
3   background-color: yellow;
4   margin: 0px;
5   padding: 5px;
6   border-top: 2px solid black;
7   font-family: Arial;
8 }
```

La figura 18 muestra el pie de página correspondiente al diseño planteado.

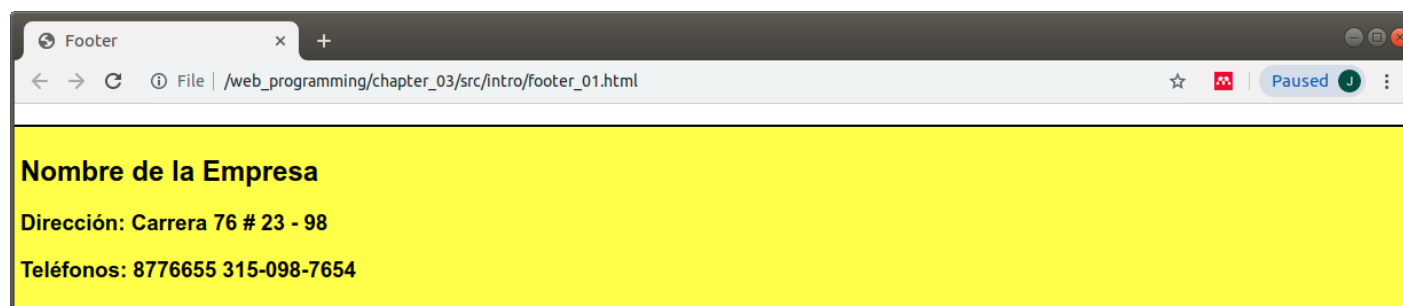


Figure 18: Pie de página (Footer)

10 CSS del proyecto

El código CSS correspondiente a la implementación del proyecto planteado para el capítulo se presenta en el listado 16.

Listing 16: `/src/estructura/css/estilo_01.css`(CSS del proyecto)

```
1 body {
2   margin: 0px;
3 }
4
5 /* cabezote */
6 header {
7   background-color: yellow;
8   font-family: Arial;
9   padding: 0px;
10  margin: 0px;
11 }
12 header h1 {
13   margin: 0px;
14   padding: 10px;
15   font-size: 250%;
16   text-align: middle;
17 }
```

```
18
19 /* menu horizontal */
20 .menu_hor {
21     list-style-type: none;
22     margin: 0;
23     padding: 0;
24     overflow: hidden;
25     background-color: green;
26 }
27 .menu_hor li {
28     float: left;
29     border-right: 1px solid yellow;
30 }
31 .menu_hor li a {
32     display: block;
33     color: white;
34     text-align: center;
35     padding: 12px 14px;
36     text-decoration: none;
37 }
38 .menu_hor li:last-child {
39     border-right: none;
40 }
41
42 /* maquetación productos y banners */
43 section {
44     width: 73%;
45     float: left;
46     margin-left: 10px;
47 }
48 aside {
49     width: 23%;
50     float: left;
51     margin-top: 4px;
52 }
53 aside img {
54     margin-left: 10px;
55 }
56 article {
57     clear: both;
58     /*border-bottom: dashed gray 1px;*/
59 }
60 article img {
61     float: left;
62     width: 100px;
63     height: 100px;
64     margin-right: 10px;
65 }
66
67 /* pie de página */
68 footer {
69     clear: both;
70     background-color: yellow;
71     margin: 0px;
72     padding: 5px;
73     border-top: 2px solid black;
74     font-family: Arial;
75 }
```

La figura 19 muestra la página web del proyecto correspondiente al diseño planteado.

Existen métodos y técnicas más avanzadas para realizar la implementación del proyecto que se cubrirán durante el desarrollo del curso.

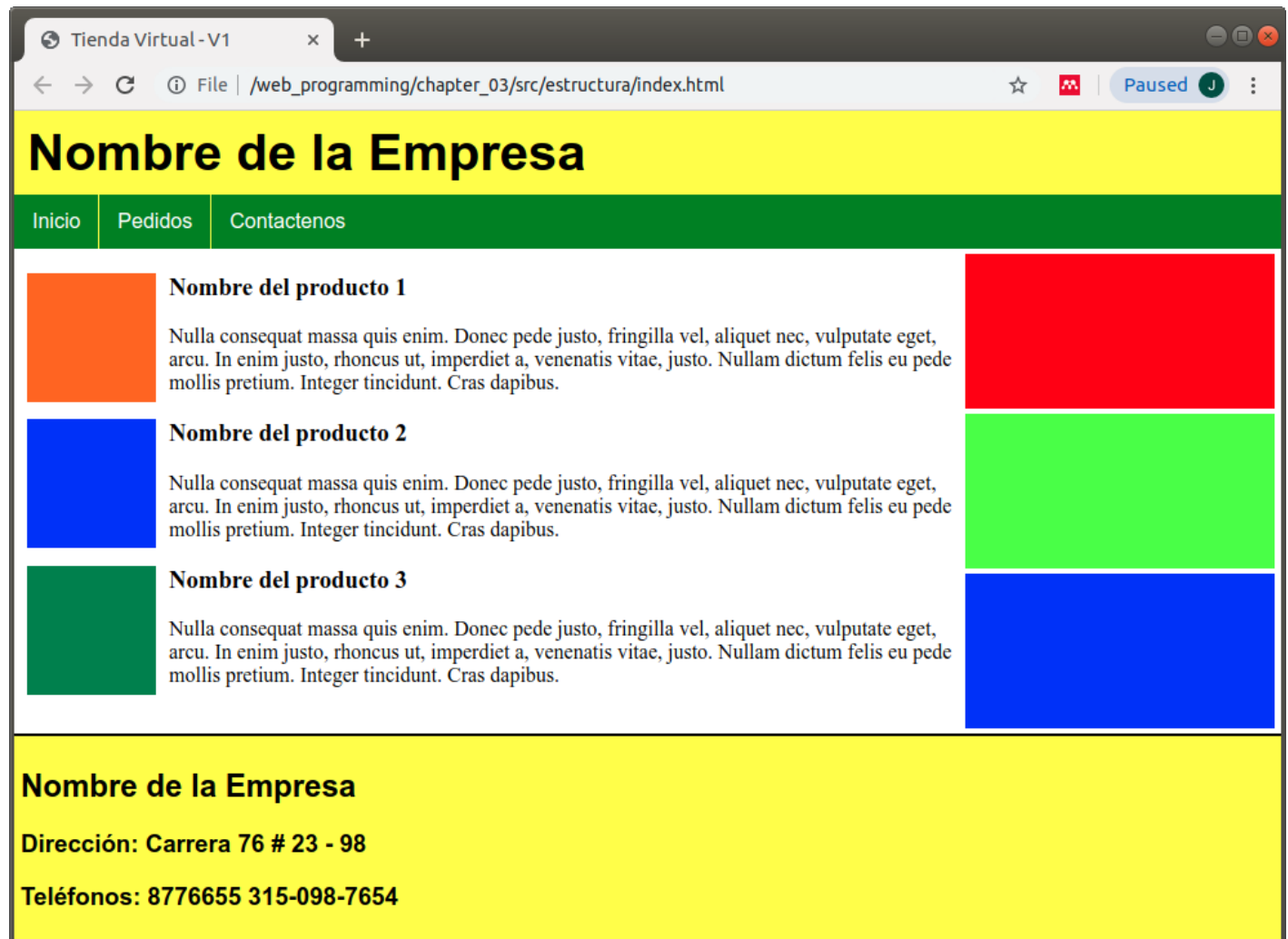


Figure 19: Página web del proyecto

11 Diseño Web Adaptable (Responsive Web Design-RWD)

En esta sección se desarrollará una primera aproximación al diseño web adaptable (Responsive Web Design), es decir una pagina web que se adapta a diferentes resoluciones de pantalla, teniendo en cuenta la gran variedad de dispositivos que pueden desplegar recursos de la web (celulares, tablets, portatiles, equipos de escritorio y televisores, entre otros).

El listado 17 contiene el documento HTML correspondiente al proyecto.

Listing 17: /src/responsive/responsive_01.html (HTML responsive)

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link rel="stylesheet" type="text/css" href="css/responsive_01.css">
5 </head>
6 <body>
7 <div class="wrapper">
8   <header class="main-head">Header</header>
9   <nav class="main-nav">
10     <ul>
11       <li><a href="">Nav 1</a></li>
12       <li><a href="">Nav 2</a></li>
13       <li><a href="">Nav 3</a></li>
14     </ul>
15   </nav>
16
17   <section class="content">
18     <h3>Section</h3>
19     <article>
20       <p>En este dise ñ o , las áreas se muestran en el orden en que están escritas en las
21         pantallas de menos de 500 píxeles de ancho.</p>
22     </article>
23     <article>
24       <p>Pasamos a un dise ñ o de dos columnas, y luego a uno de tres columnas mediante la
25         redefinición de la rejilla y la colocación de los elementos en ella.</p>
26     </article>
27   </section>
28
29   <aside class="side">Aside</aside>
30
31   <footer class="main-footer">Footer</footer>
32 </div>
</body>
</html>

```

El código CSS correspondiente a la implementación del diseño web adaptable se presenta en el listado 18.

Listing 18: /src/responsive/css/responsive_01.css (CSS responsive)

```

1 .wrapper {
2   max-width: 1024px;
3   margin: 0 auto;
4   display: grid;
5   grid-template-columns: repeat(12, [col-start] 1fr);
6   grid-gap: 2px;
7 }
8 .wrapper > * {
9   border: 2px solid #f08c00;
10  background-color: #ffec99;
11  padding: 10px;
12  grid-column: col-start / span 12;
13 }
14 nav ul {
15   list-style: none;
16   margin: 0;
17   padding: 0;
18 }
19
20 @media (min-width: 500px) {
21   .content {
22     grid-column: col-start / span 9;
23   }
24   .side {
25     grid-column: col-start 10 / span 3;

```

```
26 }
27 nav ul {
28     display: flex;
29 }
30 nav ul li {
31     margin-right: 10px;
32 }
33 }
34
35 @media (min-width: 700px) {
36     .main-nav {
37         grid-column: col-start / span 2;
38     }
39     .content {
40         grid-column: col-start 3 / span 8;
41     }
42     .side {
43         grid-column: col-start 11 / span 2;
44     }
45     nav ul {
46         flex-direction: column;
47     }
48 }
```

Para hacer las pruebas del diseño adaptable existen multiples herramientas, en esta sección vamos a utilizar del navegador Mozilla Firefox la herramienta Responsive Design Mode (Modo de diseño adaptable) que se encuentra en el menú de herramientas Web Developer (Desarrollador Web).

11.1 ancho < 500 pixels

Este proyecto se va orientar al móvil primero (mobile first), y todo lo que se quiere para las pantallas más estrechas es que los elementos permanezcan en el orden de origen, y que todos se expandan a lo largo de la cuadrícula.

Las reglas de estilo aplicadas a la clase del contenedor principal (wrapper=envoltorio) definen un ancho máximo de 1024 pixels (max-width: 1024px;), un margen de 0 unidades (pixels) y el valor "auto" centra horizontalmente el elemento dentro de su contenedor (body para este caso). El elemento tomará el ancho especificado, y el espacio restante será dividido igualmente entre los dos márgenes (margin: 0 auto;). La clase wrapper (.wrapper) tendrá una visualización (display) tipo cuadrícula (display: grid;), tendrá doce columnas (grid-template-columns: repeat(12, [col-start] 1fr;)) y una separación entre columnas de 2 pixels (grid-gap: 2px;).

Para todos los elementos que están dentro del elemento con clase wrapper, el contenedor principal, se les define por defecto un ancho de doce columnas (grid-column: col-start / span 12;), de esta manera se asume un diseño móvil primero.

Listing 19: /src/intro/css/footer_01.css (CSS del footer)

```
1 .wrapper {
2     max-width: 1024px;
3     margin: 0 auto;
4     display: grid;
5     grid-template-columns: repeat(12, [col-start] 1fr);
6     grid-gap: 2px;
7 }
8 .wrapper > * {
9     border: 2px solid #f08c00;
10    background-color: #ffec99;
11    padding: 10px;
12    grid-column: col-start / span 12;
13 }
14 nav ul {
15     list-style: none;
16     margin: 0;
17     padding: 0;
18 }
```

En la figura 20 se muestra que todos los elementos ocupan un ancho de doce columnas y aparecen uno debajo del otro en el orden en que fueron creados en la página web (header, nav, section, aside, footer).

La figura 21 muestra la página web adaptada a una resolución de pantalla menor que 500 pixels (ancho < 500 pixels).

Header
Nav 1 Nav 2 Nav 3
Section En este diseño, las áreas se muestran en el orden en que están escritas en las pantallas de menos de 500 píxeles de ancho. Pasamos a un diseño de dos columnas, y luego a uno de tres columnas mediante la redefinición de la rejilla y la colocación de los elementos en ella.
Aside
Footer

Figure 20: Diseño adaptable - responsivo (ancho < 500 pixels)

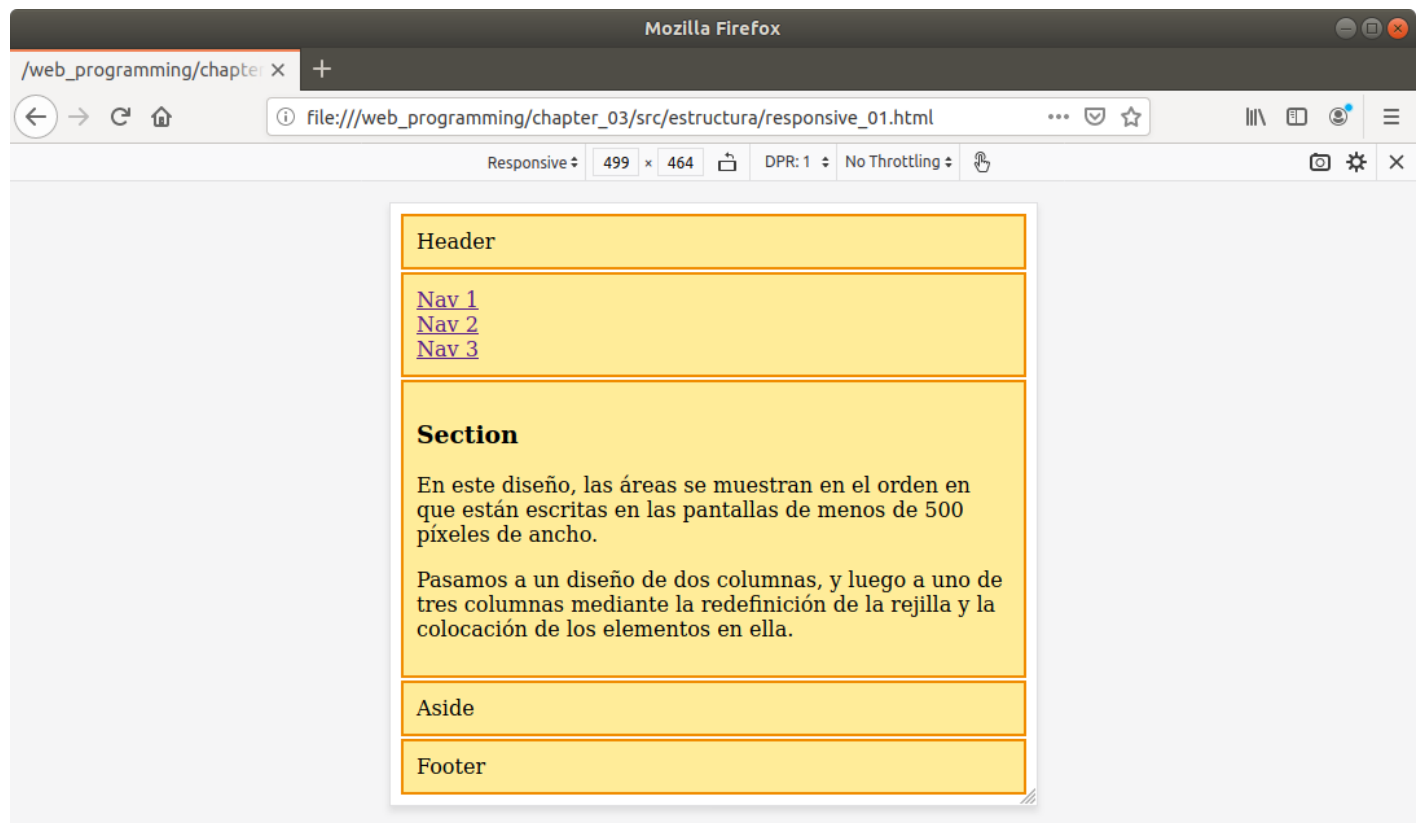


Figure 21: Página web adaptable (ancho < 500 pixels)

11.2 ancho ≥ 500 y < 700 pixels

Las reglas de estilo definen un media query (consulta de medios) que chequea el ancho del viewport (área de despliegue de la página web), si se cumple que el ancho es mínimo de 500 pixels (@media (min-width: 500px) {...}), se aplican los estilos definidos dentro de las llaves, como un condicional.

El elemento con la clase content (section para este caso) se ubicará en la columna 1 (col-start) y tendrá un ancho de 9 columnas (span 9), (.content {grid-column: col-start / span 9;}), el elemento con la clase side (aside para este ejemplo) se ubicará en la columna 10 (col-start 10) y tendrá un ancho de 3 columnas (span 3), (.side {grid-column: col-start 10 / span 3;}).

Los media queries son útiles cuando se desea modificar una página web o aplicación en función del tipo de dispositivo (como una impresora o una pantalla) o de características y parámetros específicos (como la resolución de la pantalla o el ancho del viewport del navegador).

La orientación de los elementos de lista de la barra de menú se cambia para que aparezcan uno al lado del otro (nav ul {display: flex;}) y se asigna un margen a la derecha de 10 pixels para separarlos (nav ul li {margin-right: 10px;}).

```

20 @media (min-width: 500px) {
21   .content {
22     grid-column: col-start / span 9;
23   }
24   .side {
25     grid-column: col-start 10 / span 3;
26   }
27   nav ul {
28     display: flex;
29   }
30   nav ul li {
31     margin-right: 10px;
32   }
33 }
```

En la figura 22 se muestra que los elementos header, nav y footer ocupan un ancho de doce columnas. Los elementos section y aside aparecen en una misma fila y ocupan un ancho de 9 y 3 columnas respectivamente.

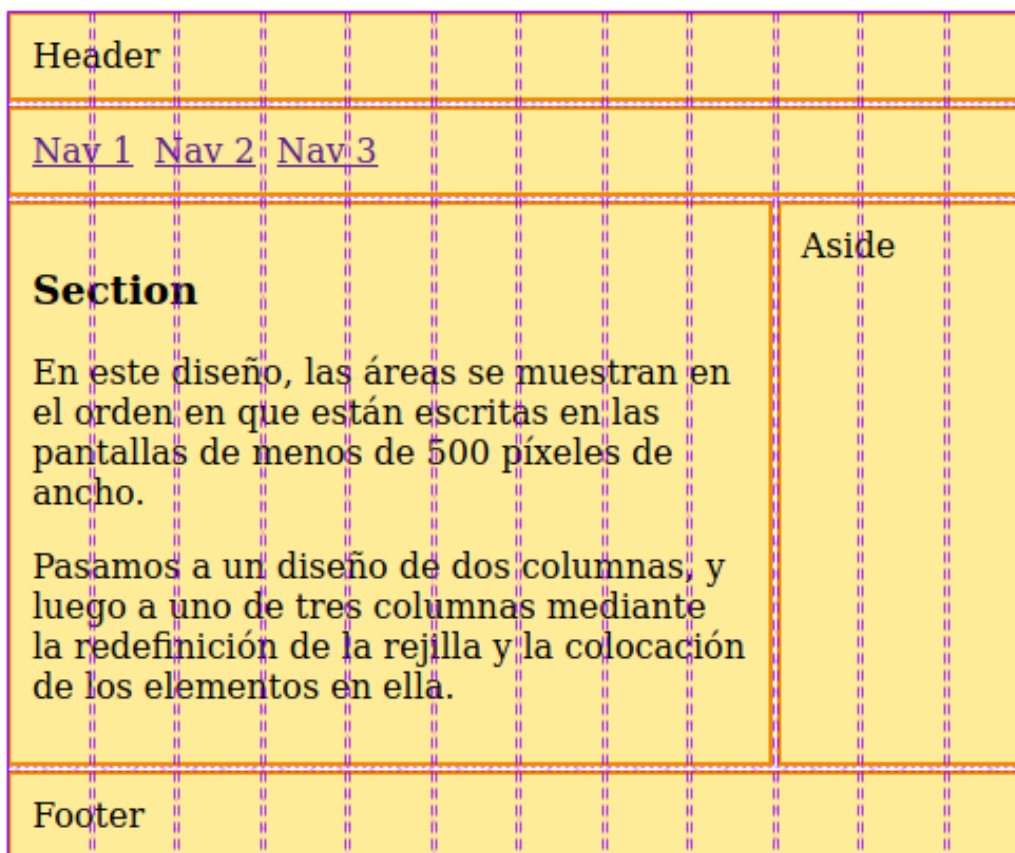


Figure 22: Diseño adaptable - responsivo (ancho ≥ 500 y < 700 pixels)

La figura 23 muestra la página web del proyecto correspondiente al diseño planteado.

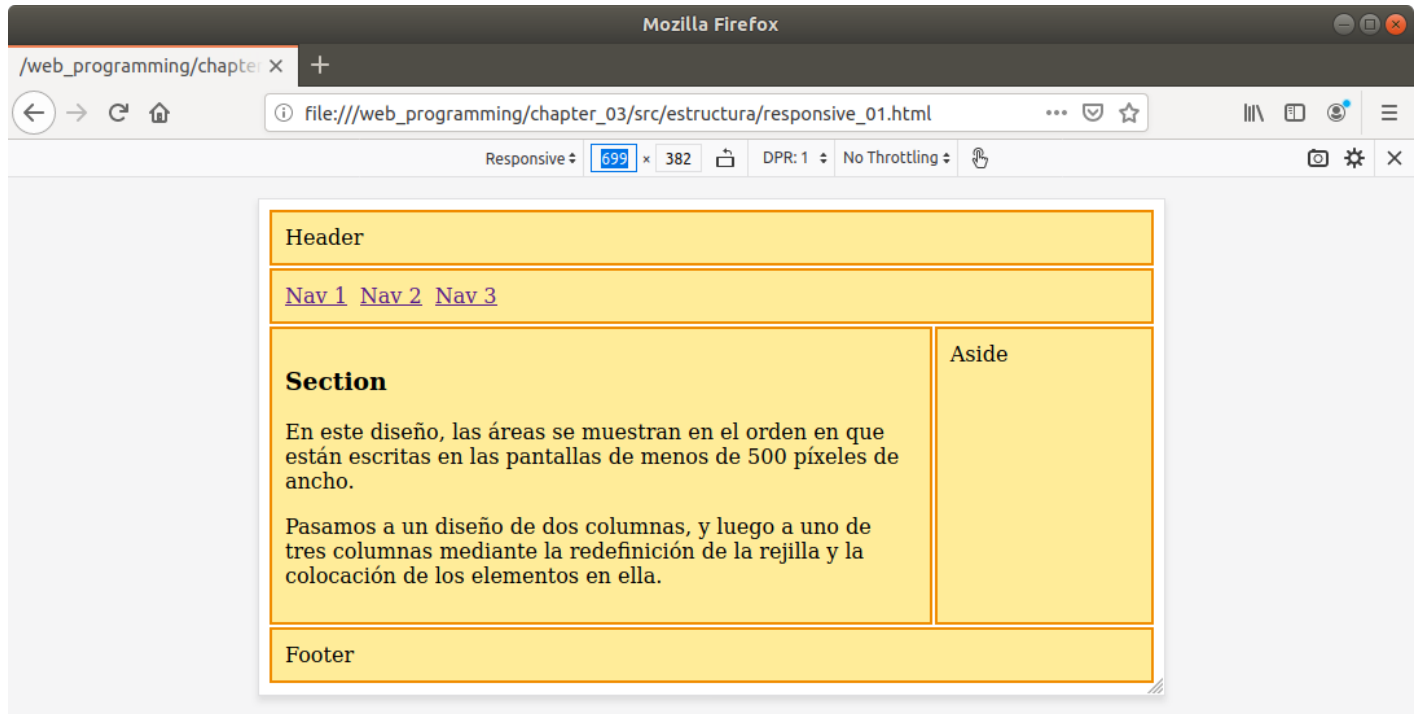


Figure 23: Página web adaptable (ancho ≥ 500 y < 700 pixels)

11.3 ancho \geq 700 pixels

Se define un media query que chequea el ancho del viewport, si el ancho es mínimo de 700 pixels (`@media (min-width: 700px) {...}`), se reorganizan en una sola fila los elementos `nav`, `section` y `aside`. El elemento con la clase `main-nav` (`nav`) se ubicará en la columna 1 y tendrá un ancho de 2 columnas (`.main-nav {grid-column: col-start / span 2;}`), el elemento con la clase `content` (`section`) se ubicará en la columna 3 y ocupará un ancho de 8 columnas (`.content {grid-column: col-start 3 / span 8;}`), el elemento con la clase `side` (`aside`) se posicionará en la columna 11 y tendrá un ancho de 2 columnas (`.side {grid-column: col-start 11 / span 2;}`) y el elemento `ul` de la barra de navegación tendrá una distribución en columnas (`nav ul {flex-direction: column;}`), un elemento de lista debajo del otro.

```

35 @media (min-width: 700px) {
36   .main-nav {
37     grid-column: col-start / span 2;
38   }
39   .content {
40     grid-column: col-start 3 / span 8;
41   }
42   .side {
43     grid-column: col-start 11 / span 2;
44   }
45   nav ul {
46     flex-direction: column;
47   }
48 }

```

La figura 24 representa el diseño para pantallas con un ancho mayor o igual a 700 pixels, muestra la ubicación de los componentes y el número de columnas que ocupan.

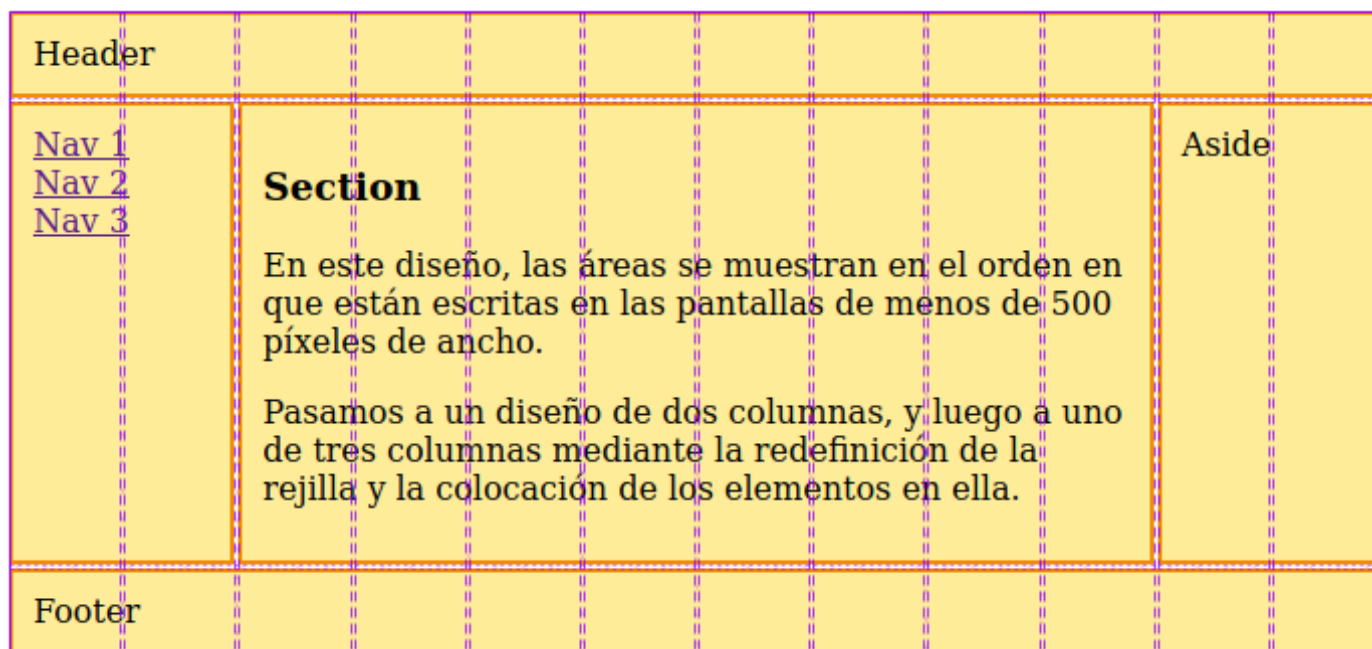


Figure 24: Diseño adaptable - responsivo (ancho \geq 700 pixels)

La figura 25 muestra la página web del proyecto. Es importante resaltar que el ancho máximo del contenedor principal es 1024 pixels y que en anchos de pantalla mayores estará centrado, debido a la aplicación de las reglas de estilo (`max-width: 1024px; margin: 0 auto;`).

12 Retos - Challenges

Convertir el proyecto de este capítulo (Tienda Virtual) a un diseño web adaptable que responda a tres anchos de pantalla diferentes:

1. Pequeño (ancho $<$ 480px) con un diseño de una columna.

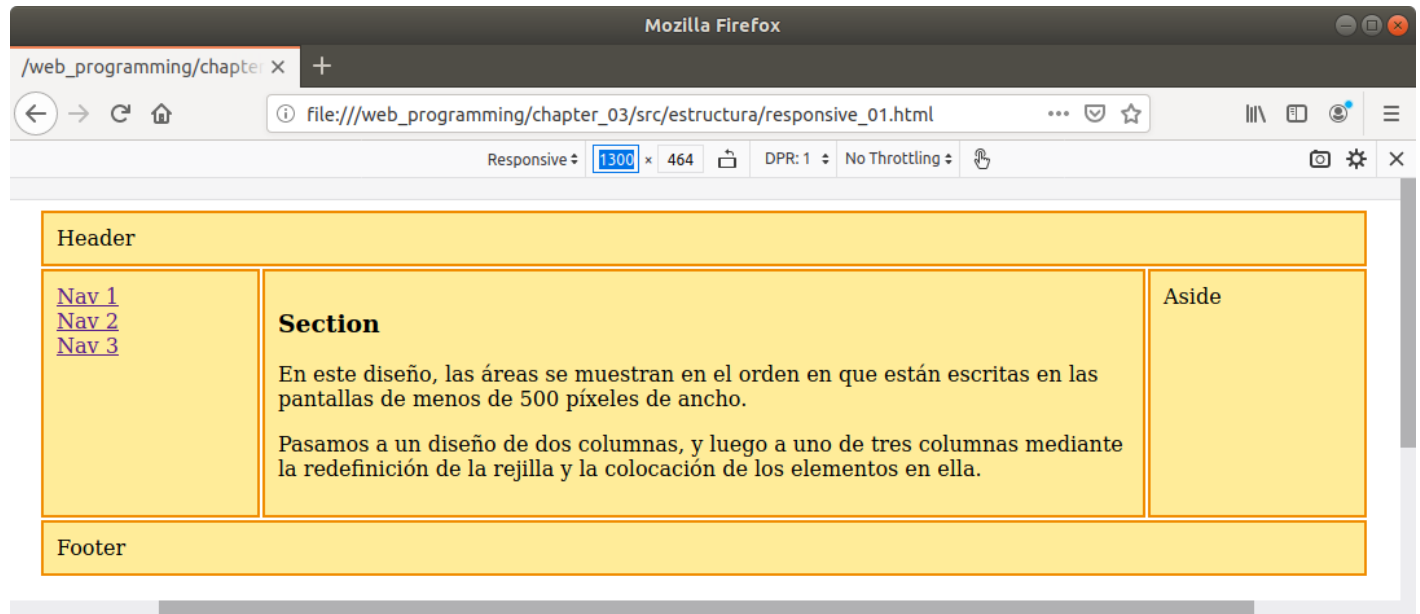


Figure 25: Página web adaptable (ancho ≥ 700 pixels)

2. Mediano (ancho $\geq 480\text{px}$ y $< 800\text{px}$) con un diseño de dos columnas.
3. Grande (ancho $> 800\text{px}$) con un diseño de tres columnas.

NOTA: El número de columnas no hace referencia a la disposición en cuadrícula (grid), sino al número de elementos contenedores que se muestran en una fila, al diseño de la página en general (Layout).