

Confiabilidad, pruebas y análisis de la información

Ejercicio de referencia para la creación de una plantilla



Plantilla

La plantilla que se va a crear tendrá todos los elementos HTML que componen la estructura vista y permitirá conocer el paso a paso de cómo se hace un proceso de maquetación siguiendo el modelo de caja tradicional.

Como se seguirá el modelo de cajas tradicional, este modelo necesita agrupar todas las cajas en una para poder centrarla y darle un tamaño específico, por eso se agregará una etiqueta <div> que agrupará todos los elementos de la página.

Con base en lo anterior la plantilla queda de la siguiente manera:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="description" content="Plantilla HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Mi primera maquetación</title>
  <link rel="stylesheet" type="text/css" href="CSS/estilos.css">
</head>
<body>

<div id="contenedor">
  <header id="cabecera">
    <h1>Desarrollo web con HTML5</h1>
  </header>
  <nav id="menu">
    <ul>
      <li>Inicio</li>
      <li>Servicios</li>
      <li>Proyectos</li>
      <li>Contacto</li>
    </ul>
  </nav>
  <section id="seccion">
    <article>
      <header>
        <hgroup>
          <h1>Revolución tecnológica 4.0</h1>
          <h2>Internet de las cosas</h2>
        </hgroup>
      </header>
    </article>
  </section>
</div>
```

```

        <time datetime="2020-09-15" pubdate>publicado
        15-09-2020|
        </time>
    </header>
    Este parrafo es de prueba para el articulo de internet de
    las cosas, por eso se tomara un parrafo ejemplo.
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim
    ad minim veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex
    ea commodo
    consequat. Duis aute irure dolor in reprehenderit in
    voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur sint
    occaecat cupidatat non
    proident, sunt in culpa qui officia deserunt mollit anim id
    est laborum.
    <figure>
        
    </figure>
    <figure>
        
        <figcaption>
            Imagen de IOT
        </figcaption>
    </figure>
    <footer>
        <p>comentarios (0)</p>
    </footer>
</article>

```



```
<article>
  <header>
    <hgroup>
      <h1>Revolución tecnológica 4.0</h1>
      <h2>Ejemplo de otro articulo</h2>
    </hgroup>
    <time datetime="2020-09-15" pubdate>publicado
      10-09-2020
    </time>
  </header>
```

Este parrafo es de prueba para el articulo de internet de las cosas, por eso se tomara un parrafo ejemplo. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse

consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
<figure>
  
  <figcaption>
    Imagen de IOT para el otro articulo
  </figcaption>
</figure>
<footer>
  <p>comentarios (0)</p>
</footer>
</article>
</section>
```

```

</section>
<aside id="columna">
    <blockquote>Texto Mensaje uno</blockquote>
    <blockquote>Texto Mensaje dos</blockquote>
</aside>
<footer id="pie">
Derechos Reservados &copy; Jose Aguilar – 2020
</footer>
</div>
</body>
</html>

```

Ilustración 50. Plantilla HTML

Fuente: Aguilar, J. 2020.

Como se puede observar la plantilla de la imagen 8 contiene todos los elementos de HTML5 (section, header, aside, article, footer) que permiten darle un esqueleto y sentido a la página para el entendimiento del navegador.

Así como está la plantilla si se revisa en un navegador se ve de la siguiente manera:



Ilustración 54. Plantilla son márgenes y padding en cero

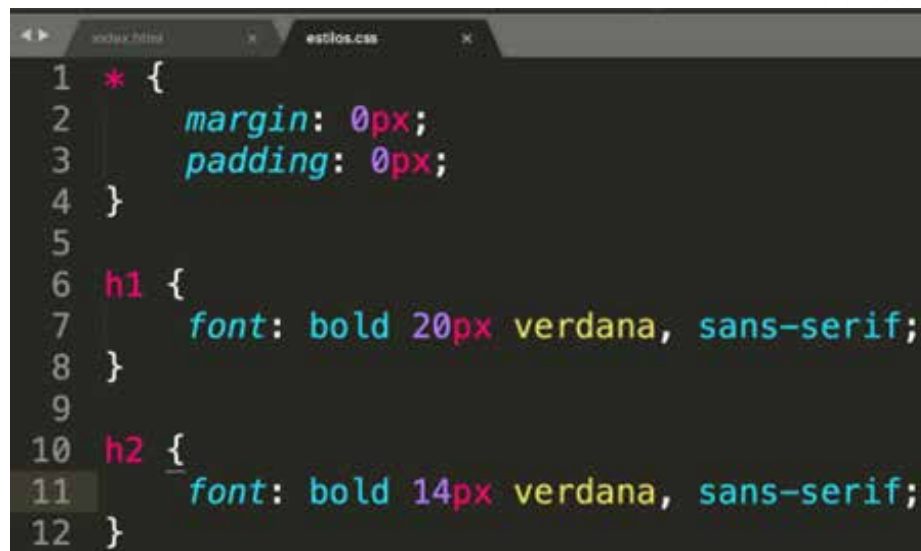
Fuente: Aguilar, J. 2020

La imagen anterior muestra que en el navegador los elementos ya no tienen ninguna margen y se pegan totalmente a los bordes de la página, de esta manera ya están listos para aplicarle las reglas necesarias sin que sus valores que tenían por defecto influyan alterando el resultado.

Con el margin y el padding en cero se asegura que todos los elementos tengan sus márgenes interno y externo en cero pixeles.

Para continuar con las reglas se van a personalizar los títulos y subtítulos que se utilizan en la página, es decir, se va a cambiar el tamaño, tipo de letra y la fuerza o el grueso de la letra.

Para que estas reglas se apliquen a todos los títulos y subtítulo de h1 y h2 se debe utilizar la aplicación de estilo por referencia, es decir, referenciando directamente la etiqueta.



```

1 * {
2   margin: 0px;
3   padding: 0px;
4 }
5
6 h1 {
7   font: bold 20px verdana, sans-serif;
8 }
9
10 h2 {
11   font: bold 14px verdana, sans-serif;
12 }

```

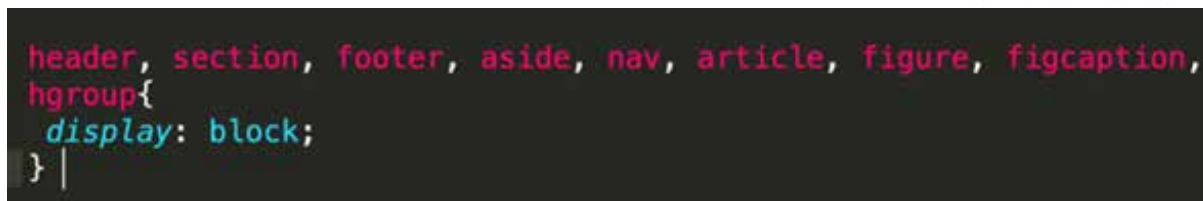
Ilustración 55. Estilos en etiquetas h1 y h2

Fuente: Aguilar, J. 2020

La imagen anterior muestra la propiedad font, asignada a los elementos <h1> y <h2> la cual permite aplicar todos los estilos para los títulos y subtítulos en una sola línea.

Las propiedades que se aplican usando la etiqueta font son: el primer valor es el de font-style donde se está diciendo que es estilo bold (negrita), el siguiente valor es font-size y se asigna el tamaño de la fuente, para terminar con la font-family que sería la familia o tipo de fuente que se utilizará para la letra.

El paso siguiente es darles una regla básica a los elementos nuevos de HTML5 porque algunos navegadores aún no los identifican y los tratan como inline. Estos elementos se necesitan de tipo block para poder aplicar el estilo de cajas tradicional.



```

header, section, footer, aside, nav, article, figure, figcaption,
hgroup{
  display: block;
} |

```

Ilustración 56. Estilo block a elementos HTML5

Fuente: Aguilar, J. 2020

La imagen anterior muestra cómo se aplica una regla display, esta regla indica cómo entiende el navegador los elementos y el tipo block es colocar como si fuera cajas cada elemento y los pondría uno debajo del otro.

Continuando con el modelo de caja tradicional el primer elemento a personalizar o impactar con reglas CSS es el body. A esta etiqueta se le debe aplicar un tamaño máximo para que se acomode a las diferentes pantallas y se debe posicionar horizontalmente.

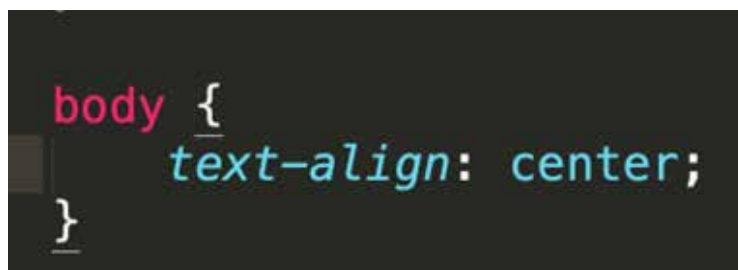


Ilustración 57. Centrando el contenido del cuerpo
Fuente: Aguilar, J. 2020



Ilustración 58. Página con el texto del body centrado.
Fuente: Aguilar, J. 2020

La imagen anterior muestra como el navegador comienza a visualizar el contenido acorde a los estilos que se van aplicando.

Siguiendo con la maquetación ahora se le debe dar un tamaño máximo al cuerpo de la página y esto se hace a través del elemento div que contiene todos los elementos del cuerpo de la página.

Este div será la caja principal del modelo de caja que se está construyendo, de esta forma se modificará el tamaño y alineación del elemento.



Ilustración 59. Estilo al div contenedor
Fuente: Aguilar, J. 2020

Como se observa en el código anterior se está haciendo referencia por el id del elemento que en este caso se llama contenedor.

Al contenedor que es el div que encierra todos los elementos de la página se le aplican reglas de tamaño de ancho, alineación de contenido o texto a la izquierda y el margin indica que da 15px al margen superior e inferior del div y se declara como automático los márgenes de izquierda y derecha.

A continuación, se muestra cómo se ven estos estilos aplicados en el navegador.



Ilustración 60. Estilo de contenedor desde navegador.

Fuente: Aguilar, J. 2020

En la imagen anterior se ve como el contenedor ajusta el tamaño del contenido a un ancho específico y queda centrado en la página porque el body tiene el estilo text-align que centra su contenido.

Continuando con la maquetación se debe seguir con la etiqueta <header>, recuerde que esta etiqueta se le asignó un id llamado cabecera.

Como se mencionó al inicio cada elemento de HTML5 es interpretado como block y también el CSS inicial asegura ese bosquejo, un elemento block es una caja que verticalmente ocupa el tamaño según su contenido y horizontalmente ocupa el 100% de la página o de su elemento padre.

Teniendo esto presente se aplican los siguientes estilos a la cabecera.

```
#contenedor {
  width: 960px;
  margin: 15px auto;
  text-align: left;
}

#cabecera {
  background: #FFFBB9;
  border: 1px solid #999999;
  padding: 20px;
}
```

Ilustración 61. Estilos de la cabecera

Fuente: Aguilar, J. 2020

El código que se aplica para personalizar la cabecera indica un código hexadecimal que asigna un color amarillo a través de la regla background, también en la regla border se crea un borde con una línea sólida de 1px de grosor y con color gris como indica el código hexadecimal.

Los colores hexadecimales los puede revisar y tomar de la siguiente página:

<https://htmlcolorcodes.com/es/>

El padding aplica un margen interno de 20px en el elemento que contiene la cabecera. Lo anterior se visualiza en el navegador de la siguiente manera.



Ilustración 62. Cabecera personalizada con CSS

Fuente: Aguilar, J. 2020

El navegador muestra la cabecera ajustada con los colores y el borde aplicados en los códigos CSS. Ahora se va a ajustar la barra de navegación que será el menú de la página, para ponerle un estilo horizontal y adecuado según las necesidades.

```
#menu {
  background: #CCCCCC;
  padding: 5px 15px;
}

#menu li {
  display: inline-block;
  list-style: none;
  padding: 5px;
  font: bold 14px verdana, sans-serif;
}
```

Ilustración 63. Estilos para la barra de navegación

Fuente: Aguilar, J. 2020

El código anterior se aplica al id llamado menú que sería el que se puso a la etiqueta HTML <nav>, se da un estilo de color de fondo gris y unos márgenes internos de 5px y 15px.

La propiedad padding funciona lo mismo que margin, tiene cuatro valores los cuales son: superior, derecho, inferior, izquierdo, en este orden. Si se declara solo un valor, este será asignado para todos los espacios alrededor del contenido del elemento. En cambio, si se especifican dos valores, el primero será asignado como margen interno de la parte superior e inferior del contenido y el segundo valor será asignado al margen interno de los lados, izquierdo y derecho.

El siguiente código que tiene la referencia #menu li eso indica que se van aplicar los estilos al elemento li que contenga como elemento padre uno con id llamado menú.

Las etiquetas li en la plantilla diseñada es la que tiene los ítems del menú y por eso se le aplicaran las propiedades de display y list-style. Display indica cómo se va a visualizar los elementos y será en inline-block diciendo que serán bloques, pero en línea horizontal.

La propiedad list-style tiene un valor de none, esto indica que no tenga ningún estilo de lista y así se quitan las viñetas. El padding y font son propiedades que ya se han aplicado y conocen su funcionamiento.



Ilustración 64. Visualización barra de navegación.

Fuente: Aguilar, J. 2020

Como se observa el menú ya tiene forma y se ve como las barras de navegación encontrada en las diferentes páginas. Una vez configurado el menú a través de estilos se procede con los siguientes elementos en la estructura que son el section y aside.

Siguiendo con el modelo de caja tradicional al section y al aside se le aplicara la propiedad float donde el bloque se vuelve flotante dentro del espacio de la página donde esta y se puede ubicar a la izquierda o derecha según las necesidades del diseño.

```
#seccion {
  float: left;
  width: 660px;
  margin: 20px;
}

#columna {
  float: left;
  width: 220px;
  margin: 20px 0px;
  padding: 20px;
  background: #CCCCCC;
}
```

Ilustración 65. Propiedades aplicadas a etiquetas section y aside.

Fuente: Aguilar, J. 2020

El Código muestra como al id sección que corresponde a la etiqueta section se aplica el float para que ese bloque flote a la izquierda del espacio de pantalla en donde esta y ocupando un ancho (width) de 660px con márgenes de 20px en todos sus lados.

Al ocupar solo 660px queda el resto de campo para que floten los demás bloques necesarios, en este caso el siguiente bloque es el aside que tiene el id llamado columna. Esta etiqueta es personalizada con un float de valor izquierdo, un ancho de 220px con unos márgenes internos y externos específicos, así como un color de fondo gris.



Ilustración 66. Estilos en section y aside.

Fuente: Aguilar, J. 2020

En el navegador se puede observar como el section se sitúa en la parte izquierda de la pantalla y como es flotante permite acoplar la caja del aside también flotante del lado izquierdo del espacio que queda. Estas secciones tienen estilo de ancho definido, fuente de letra y el aside un color de fondo.

Para terminar la estructura principal se modificará el elemento estructural faltante y es el pie de página (footer).

Al footer se le aplican los siguientes estilos.

```
#columna {
    float: left;
    width: 220px;
    margin: 20px 0px;
    padding: 20px;
    background: #CCCCCC;
}

#pie {
    clear: both;
    text-align: center;
    padding: 20px;
    border-top: 2px solid #999999;
}
```

Ilustración 67. Estilos del footer.

Fuente: Aguilar, J. 2020

La propiedad clear restaura el flujo normal del documento, es decir, prohíbe que el elemento en cuestión se posicione al lado izquierdo o derecho de alguna caja flotante.

El valor **both** significa que ambos lados del elemento se restaurarán al flujo normal de bloques uno debajo de otro.

Clear tiene también como función empujar los bloques verticalmente, haciendo que los bloques flotantes ocupen un área real en la pantalla. Sin esta propiedad, el navegador presenta el documento en pantalla como si los elementos flotantes no existieran y las cajas se superponen.



Imágenes de IOT para el otro artículo
comentarios (0)

Derrochos Reservados © José Aguilar - 2020

Ilustración 68. Visualización footer

Fuente: Aguilar, J. 2020

El navegador muestra cómo queda el footer debajo de los elementos section y aside maquetando la forma de la página para terminar la estructura base.

A continuación, se muestra como es el modelo de cajas tradicional que acabamos de aplicar.

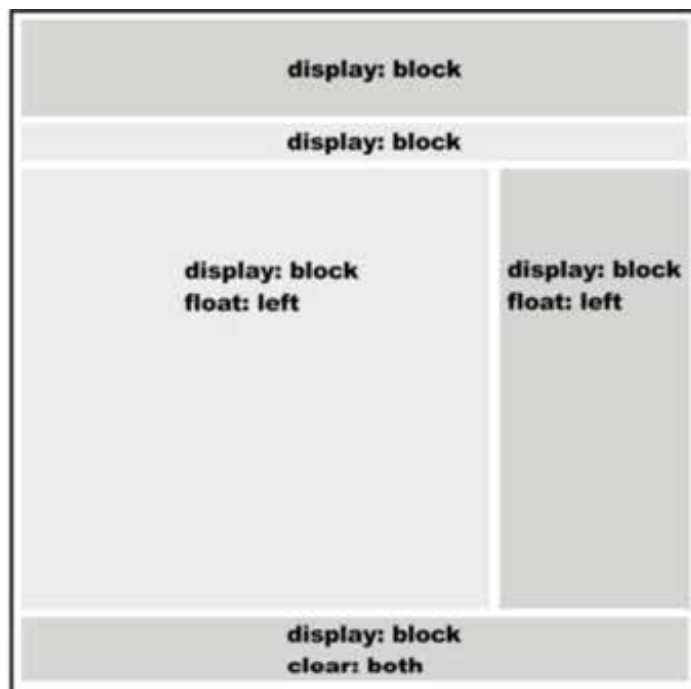


Ilustración 69. Representación del modelo de cajas tradicional

Fuente: Aguilar, J. 2020

Ahora para terminar la maquetación se van a dar estilos a los artículos o los elementos article y las imágenes.

Primero se personalizarán las imágenes en su tamaño y su título para que se ajusten a lo necesitado a través del siguiente código:

```
img{
  width: 70%;
  height: 70%;
}

figcaption {
  font: italic 14px verdana, sans-serif;
}
```

Ilustración 70. Estilos en las imágenes de la página.

Fuente: Aguilar, J. 2020

Con esas propiedades aplicadas se le está diciendo a la imagen que su tamaño en ancho y alto será el 70% del original y acorde al tamaño de la página y el espacio en donde este contenida.

Las otras propiedades cambian la fuente del título de la imagen.

Ahora se terminará de maquetar esta plantilla dándole un estilo a los artículos.

```

article {
  background: #FFFBCC;
  border: 1px solid #999999;
  padding: 20px;
  margin-bottom: 15px;
}

article footer {
  text-align: right;
}

time {
  color: #999999;
}

```

Ilustración 71. Estilos etiqueta <article>

Fuente: Aguilar, J. 2020

Los estilos que aparecen en la imagen anterior ya son conocidos y son los que se han estado utilizando a lo largo de esta maquetación, color de fondo, borde, márgenes internos y externos, alineación del pie de página a la derecha y color gris del texto de publicación del artículo. Con estos estilos se termina la maquetación y la página se ve de la siguiente manera.



Ilustración 72. Visualización plantilla maquetada.

Fuente: Aguilar, J. 2020

Así, con este modelo de cajas tradicional se puede hacer cualquier organización o diseño que se desee, en este caso se hizo este ejemplo como practica dentro del proceso de adquisición del conocimiento. Ahora se debe proceder a indicar como se hace que esa web maquetada o cualquier web que se desarrolle se adapte a las diferentes pantallas de los dispositivos.

1. Diseño Web Responsive

El diseño responsivo es como se dijo en el párrafo anterior, una página que se adapte a la pantalla de diferentes dispositivos.

Para este proceso se utiliza el estilo de sistemas de filas y columnas el cual se explica en el Anexo_1_ SistemaFilas_y_Columnas.

Ahora teniendo los recursos se comenzará a trabajar con la plantilla y se va a ir creando así la página responsiva.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
  <meta name="autor" content="Tutorial Web Responsive">
  <meta name="description" content="Esta es la descripción
  del sitio">
  <meta name="keywords" content="Tutorial, html, css">

  <title>Web Responsive</title>

  <link rel="icon" href="images/img01.jpg">

  <link href='https://fonts.googleapis.com/
  css?family=Kaushan+Script' rel='stylesheet' type='text/css'>

  <link href="CSS/estilo.css" type="text/css" rel="stylesheet"
  >
</head>
```

Ilustración 73. Head de la página

Fuente: Aguilar, J. 2020

Como se observa en la imagen anterior se tiene el código HTML del head de la página donde se asignan las diferentes etiquetas meta, el meta charset para la codificación de caracteres que va a manejar.

Ahora se crea la hoja de estilos CSS para ir en paralelo aplicando las reglas y así organizando la página. La idea del CSS es que tenga todo el sistema de columnas con las consultas. Como se tienen los archivos de columnas y consultas, se podría copiar y pegar el texto, pero para no tener que buscar los archivos, si los tenemos en otros proyectos o en otro folder, para eso se va a aprovechar un recurso que tiene sublime text para desarrolladores que se llama Snippet.

Los Snippet son atajos que se pueden crear en el sublime text. A continuación, se puede ver como se crea un Snippet.

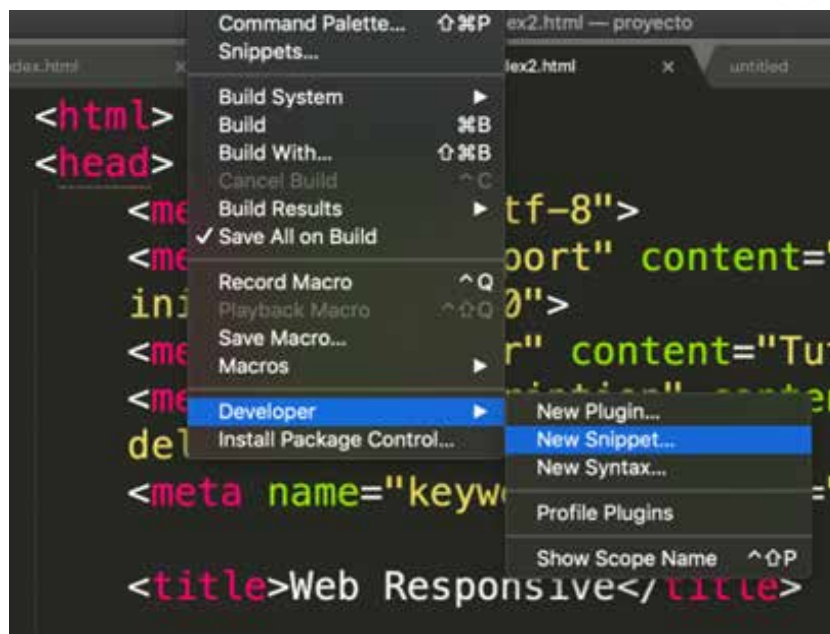


Ilustración 74. Creando Snippet
Fuente: Aguilar, J. 2020

Al crearlo muestra este código de ejemplo.

```

1 <snippet>
2   <content><![CDATA[
3   Hello, ${1:this} is a ${2:snippet}.
4   ]]></content>
5   <!-- Optional: Set a tabTrigger to define how to trigger the
6   snippet -->
7   <!-- <tabTrigger>hello</tabTrigger> -->
8   <!-- Optional: Set a scope to limit where the snippet will
9   trigger -->
10  <!-- <scope>source.python</scope> -->
11 </snippet>

```

Ilustración 75. Código Snippet
Fuente: Aguilar, J. 2020

En la imagen anterior se puede ver que el código que está en la línea 3 muestra lo que es un snippet. Ahí es donde se pueden crear variables, pero en este caso no se van a crear variable, sino que se va a pegar todo el código de columnas con consultas.


```

111 .col-xs-9{width:75%;}
112 .col-xs-8{width:66.66666667%;}
113 .col-xs-7{width:58.33333333%;}
114 .col-xs-6{width:50%;}
115 .col-xs-5{width:41.66666667%;}
116 .col-xs-4{width: 33.33333333%;}
117 .col-xs-3{width: 25%;}
118 .col-xs-2{width: 16.66666667%;}
119 .col-xs-1{width: 8.33333333%;}
120
121 }
122
123
124 ]]></content>
125 <!-- Optional: Set a tabTrigger to define how to trigger
the snippet -->
126 <!-- <tabTrigger>hello</tabTrigger> -->
127 <!-- Optional: Set a scope to limit where the snippet will
trigger -->
128 <!-- <scope>source.python</scope> -->
129 </snippet>

```

Ilustración 76. Código pegado de columnas con consultas.

Fuente: Aguilar, J. 2020

Después de pegar el código en el snippet que se está creando, se puede ver que en la línea 126 está el tab, es decir, que si se quita el comentario se puede hacer un snippet.

```

124 ]]></content>
125 <!-- Optional: Set a tabTrigger to defi
the snippet -->
126 <tabTrigger>responsivo</tabTrigger>
127 <!-- Optional: Set a scope to limit whe
trigger -->
128 <!-- <scope>source.python</scope> -->
129 </snippet>

```

Ilustración 77. Tab trigger del Snippet.

Fuente: Aguilar, J. 2020

Ahora se procede a guardar el snippet, para guardarlo se debe hacer con la extensión sublime-snippet, no se debe elegir ninguna ruta para guardar, el snippet se guarda automáticamente en la librería de sublime text para poderlo usar en cualquier proyecto con la palabra responsivo que fue la que se creó. Una vez creado el snippet se procede a copiar la palabra clave en el archivo estilos.css y se da tabulador para que pegue el código del snippet que se acabó de crear.

```
1
2      /*=====
3      GLOBAL
4      =====*/
5
6      *{
7          margin:0px;
8          padding:0px;
9          list-style: none;
10         text-decoration: none;
11         font-family: sans-serif;
12         box-sizing: border-box;
13     }
14
15     .contenedor{
16         position:relative;
```

Ilustración 78. Código de snippet en estilos.css
Fuente: Aguilar, J. 2020

Una recomendación en la maquetación de un sitio web responsivo es comenzar a modularizar el sitio, es decir, hacerlo, así como se hizo en la Unidad 1, que se sacaron módulos de encabezado, menú, etc. Dicho lo anterior se va a proceder a crear las secciones en el HTML.

```
19 <body>
20 <div class="contenedor">
21
22 <header class="fila">
23
24 </header>
25
26 </div>
27 </body>
```

Ilustración 79. primera fila
Fuente: Aguilar, J. 2020

Se crea un contenedor que va a tener todo el HTML de la página y según la imagen anterior se crea también el header al cual se le da una clase llamada fila.

Ahora para seguir creando las secciones del documento como en la maqueta anterior viene la etiqueta de navegación.

```
20 <div class="contenedor">
21
22 <header class="fila">
23
24 </header>
25 <nav class="fila"></nav>
26 </div>
27 </body>
28 </html>
```

Ilustración 80. Etiqueta nav de la plantilla

Fuente: Aguilar, J. 2020

Siguiendo con la maqueta se procederá a la etiqueta section.

```
20 <div class="contenedor">
21
22 <header class="fila"></header>
23
24 <nav class="fila"></nav>
25
26 <section id="seccion3" class="fila"></section>
27
28 <section id="seccion4" class="fila"></section>
29
30 <footer class="fila"></footer>
31
32 </div>
```

Ilustración 81. Secciones de la página

Fuente: Aguilar, J. 2020

En la imagen anterior se puede ver todas las secciones creadas, desde el header que ya se había mirado hasta el footer, lo cual va a contener toda la información de la página.

Con lo anterior ya se tienen todas las divisiones de la página que se va a pasar a diseño responsivo, ahora se debe asignar el contenido.

La siguiente imagen como se divide la primera etiqueta header en tres columnas, la primera columna para el logo, la siguiente para el espacio y la última para los botones que simulan los de las redes sociales.



Ilustración 82. División etiqueta header

Fuente: Aguilar, J. 2020

Para hacer esto se asigna el siguiente código.

```
<header class="fila">
  <div id="logo"></div>
  <div></div>
  <div></div>
</header>
```

Ilustración 83. Etiqueta Header

Fuente: Aguilar, J. 2020

Así como la imagen anterior quedaría el código, lo que falta es empezar a insertar el contenido.

Ahora se va a comenzar a dividir cada caja o div en columnas, para que ocupen las 12 columnas de las que se compone una página.

```
22 <header class="fila">
23
24   <div id="logo" class="col-lg-2">
25     Logotipo
26   </div>
27
28   <div class="col-lg-8"></div>
29
30   <div class="col-lg-2"></div>
31
32 </header>
```

Ilustración 84. División del header en Columnas

Fuente: Aguilar, J. 2020

Con lo anterior, el header queda dividido en cajas y cada una ocupa un número de columnas específicas en la pantalla, para así poder manipularla más adelante.

Lo que sigue ahora es tomar el div que ocupa 2 columnas para dividirlo en tres columnas de igual tamaño para los logos de redes sociales.

```

22 <header class="fila">
23
24   <div id="logo" class="col-lg-2">
25     Logotipo
26   </div>
27
28   <div class="col-lg-8"></div>
29
30   <div class="col-lg-2">
31     <div class="col-lg-4"></div>
32     <div class="col-lg-4"></div>
33     <div class="col-lg-4"></div>
34   </div>
35
36 </header>

```

Ilustración 85. División en columnas de redes sociales.

Fuente: Aguilar, J. 2020

La imagen anterior muestra cómo se divide el div que va a contener las redes sociales en tres div donde cada uno ocupa 4 columnas y así seguir maquetando la página.

Ahora dentro de esos div que ocupan cada uno 4 columnas se va a programar los iconos que va a tener la página, quedando de la siguiente manera.

```

30   <div class="col-lg-2">
31     <div class="col-lg-4">
32       <center id="icono1">X</center>
33     </div>
34     <div class="col-lg-4">
35       <center id="icono3">X</center>
36     </div>
37     <div class="col-lg-4">
38       <center id="icono3">X</center>
39     </div>
40   </div>

```

Ilustración 86. Columnas de iconos de redes sociales

Fuente: Aguilar, J. 2020

Lo anterior en el navegador se ve de la siguiente manera:



Ilustración 87. Vista en el navegador

Fuente: Aguilar, J. 2020

Es hora de comenzar a aplicarle estilos personalizados a esa primera sección y hacerlo de forma organizada.

Se empezará con la palabra logotipo, se le asigna el siguiente código de reglas CSS.

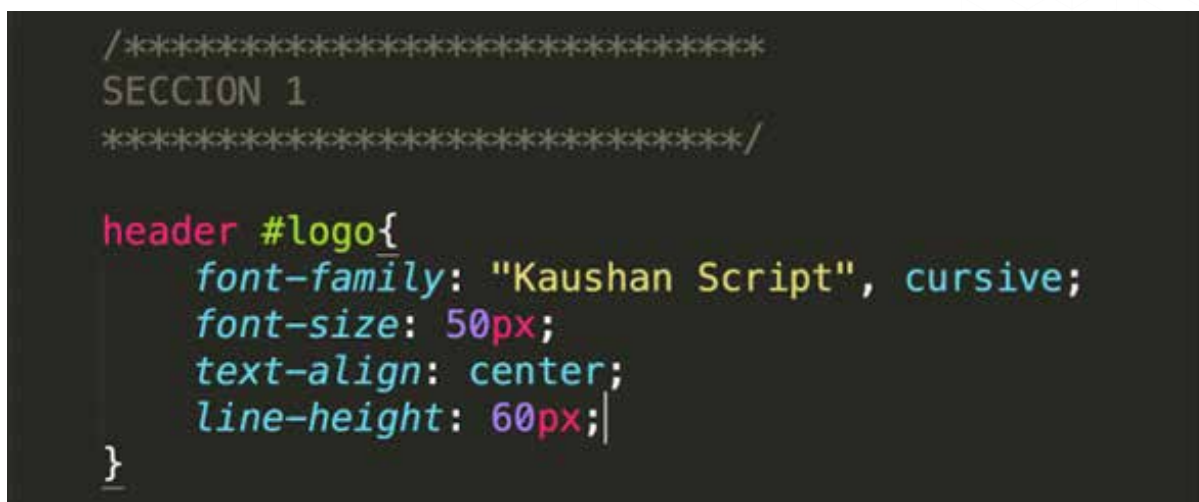


Ilustración 88. CSS para logotipo.

Fuente: Aguilar, J. 2020

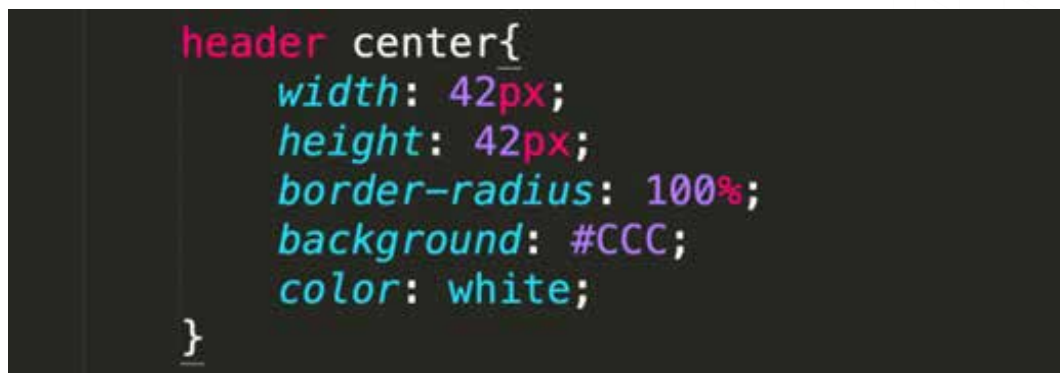
El código anterior muestra que se asigna una fuente de letra de Google, dándosele un tamaño, alineándolo al centro y dándole una altura de 60px.



Ilustración 89. Logotipo con estilo.

Fuente: Aguilar, J. 2020

Ahora se escriben las reglas para los iconos de las redes sociales, en este caso para crear la forma de botón.



```
header center#icono1{
    background: blue;
}

header center#icono2{
    background: green;
}

header center#icono3{
    background: red;
}
```

Ilustración 90. Código para las etiquetas center de los iconos.
Fuente: Aguilar, J. 2020

Aplicando el código anterior se le da un color de fondo a los iconos, se da un border radius de 100% para que queden totalmente circulares, una altura y anchura determinada y un color de letra blanco. Mostrándose en el navegador de la siguiente manera.



Ilustración 91. Iconos y logotipo en navegador.
Fuente: Aguilar, J. 2020

Ahora para que no se vean esos bordes se van a quitar desde la clase general de columnas.

```
30
31 [class*="col-"]{
32     float:left;
33     border:1px solid white;
34     padding:0px;
35 }
```

Ilustración 92. Comentando bordes.
Fuente: Aguilar, J. 2020

Para poder ocultar esos bordes se debe comentar la línea 33 de la imagen que se muestra arriba. Con lo anterior ya se puede ver la página en un dispositivo de pantalla grande, por eso el lg de las columnas, pero ahora es indispensable programar también que mantenga la estructura cuando la pantalla del dispositivo es mediana.

```

22 <header class="fila">
23
24   <div id="logo" class="col-lg-2 col-md-2">Logotipo</div>
25
26   <div class="col-lg-8 col-md-8"> </div>
27
28   <div class="col-lg-2 col-md-2">
29     <div class="col-lg-4 col-md-4">
30
31       <center id="icono1">X</center>
32
33     </div>
34     <div class="col-lg-4 col-md-4">
35
36       <center id="icono2">X</center>
37
38     </div>
39     <div class="col-lg-4 col-md-4">

```

Ilustración 93. Columnas para dispositivos de pantalla media.

Fuente: Aguilar, J. 2020

Ahora para pantallas pequeñas se debe redistribuir las columnas quedando de la siguiente manera.

```

<div id="logo" class="col-lg-2 col-md-2 col-sm-3">Logotipo</div>

<div class="col-lg-8 col-md-8 col-sm-6"> </div>

<div class="col-lg-2 col-md-2 col-sm-3">
  <div class="col-lg-4 col-md-4 col-sm-4">

    <center id="icono1">X</center>

  </div>
  <div class="col-lg-4 col-md-4 col-sm-4">

    <center id="icono2">X</center>

  </div>
  <div class="col-lg-4 col-md-4 col-sm-4">

    <center id="icono3">X</center>

```

Ilustración 94. Columnas para pantallas pequeñas.

Fuente: Aguilar, J. 2020

Revisando en el navegador en diferentes tamaños de pantalla se ve de la siguiente manera.



*Ilustración 95. Visualización en diferentes pantallas.
Fuente: Aguilar, J. 2020*

Como se observa la última imagen, se reagrupa los iconos debajo del logotipo, pero deberían quedar centrados para una mejor visualización.

Para hacer esto se agrega más información de columnas en el código HTML de la siguiente manera.

```

24 <div id="logo" class="col-lg-2 col-md-2 col-sm-3 col-xs-12">
    Logotipo</div>
25
26 <div class="col-lg-8 col-md-8 col-sm-6 col-xs-12"> </div>
27
28 <div class="col-lg-2 col-md-2 col-sm-3 col-xs-12">
29     <div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
30
31         <center id="icono1">X</center>
32
33     </div>
34     <div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
35
36         <center id="icono2">X</center>
37
38     </div>
39     <div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">

```

Ilustración 96. Se agrega el código col-xs-12

Fuente: Aguilar, J. 2020

Si se visualiza en el navegador la página se ve de la siguiente manera.



Ilustración 97. Página vista desde navegador

Fuente: Aguilar, J. 2020

Como se puede observar los botones que simulan las redes sociales no están alineados al centro, para poderlos alinear se debe agregar la letra X en una etiqueta de párrafo y su estilo aplicado se ajustaría el ya hecho para quedar así.

```

header center p{
    text-align: center;
    width: 42px;
    height: 42px;
    border-radius: 100%;
    background: #CCC;
    color: white;
    padding-top: 10px;
}

header center#icono1 p{
    background: blue;
}

header center#icono2 p{
    background: green;
}

header center#icono3 p{
    background: red;
}

```

Ilustración 98. Estilo CSS para la etiqueta p
Fuente: Aguilar, J. 2020

Con lo anterior ya realizado y que se puede visualizar que es responsivo, se va a proceder a maquetar el resto de la página.

```

<nav class="fila">
  <ul>
    <li class="col-lg-2 col-md-2 col-sm-6 col-xs-12"><a href="#">
      >Botón 1</a>
    <li class="col-lg-2 col-md-2 col-sm-6 col-xs-12"><a href="#">
      >Botón 2</a>
    <li class="col-lg-2 col-md-2 col-sm-6 col-xs-12"><a href="#">
      >Botón 3</a>
    <li class="col-lg-2 col-md-2 col-sm-6 col-xs-12"><a href="#">
      >Botón 4</a>
    <li class="col-lg-2 col-md-2 col-sm-6 col-xs-12"><a href="#">
      >Botón 5</a>
    <li class="col-lg-2 col-md-2 col-sm-6 col-xs-12"><a href="#">
      >Botón 6</a>
    </li>
  </ul>
</nav>

```

Ilustración 99. Código HTML de la barra de navegación
Fuente: Aguilar, J. 2020

Como se observa, ya se conoce cómo funcionan las columnas, cuando se está desde un dispositivo móvil “col-xs-12” se asume que cada botón ocupa el total de la pantalla, es decir, sus 12 columnas. Ahora se le da estilo a los botones.

```

/*****
SECCION 2
*****/

nav ul li{
  padding: 10px;
  text-align: center;
}

nav ul li a{
  display: block;
  background: rgba(100,0,255,1);
  font-family: "Kaushan Script", cursive;
  color: white;
  line-height: 48px;
}

```

Ilustración 100. Estilos CSS para la barra de navegación.

Fuente: Aguilar, J. 2020

El código anterior que se aplica permite personalizar la barra de navegación y ya son conocidos con por todo lo que se ha desarrollado en la Unidad anterior y lo que se lleva de esta. Lo aplicado anteriormente se ve de la siguiente manera en el navegador:



Ilustración 101. Visualización pantalla de escritorio.

Fuente: Aguilar, J. 2020

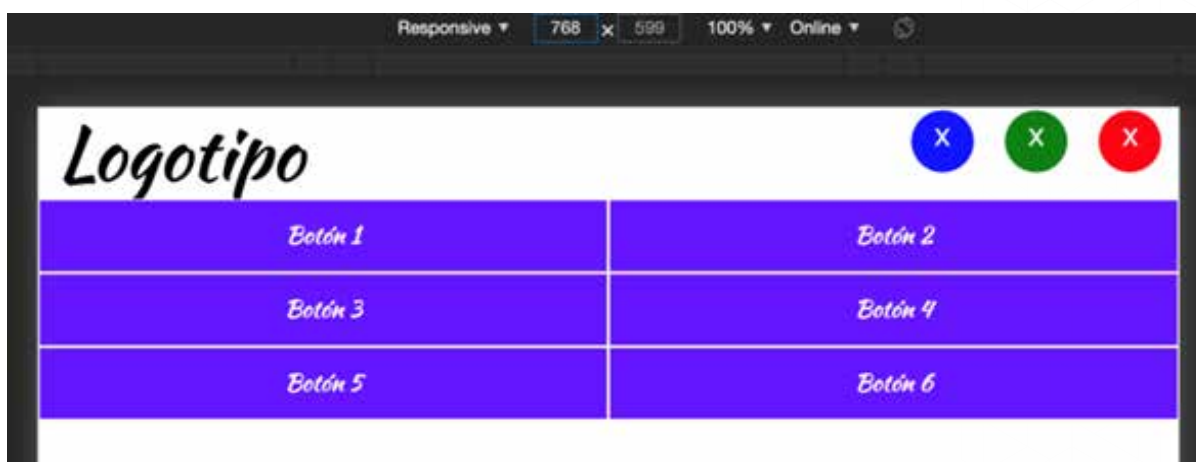


Ilustración 102. Visualización Pantalla intermedia

Fuente: Aguilar, J. 2020

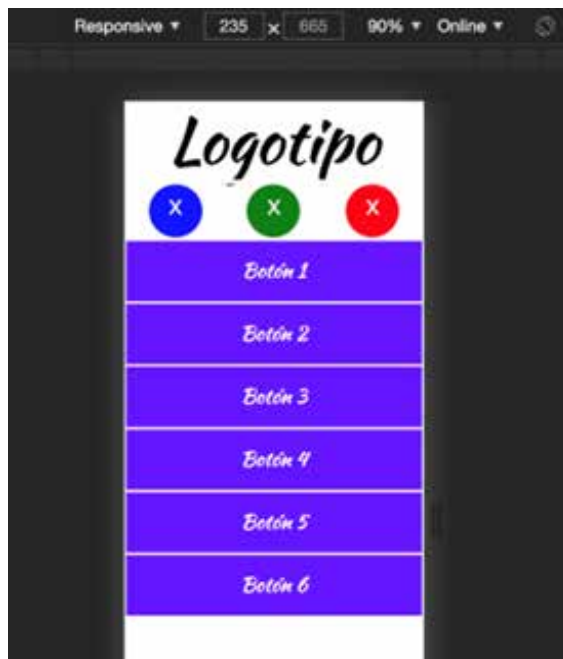


Ilustración 103. Visualización dispositivo móvil.

Fuente: Aguilar, J. 2020

Como se puede observar en las dos imágenes anteriores los botones agregados están de manera responsiva, es decir, se adaptan y organizan de acuerdo al tipo de pantalla donde se esté mostrando. Teniendo lo anterior, ahora se va a aplicar un estilo para que cuando se pasa el cursor por encima del botón, el botón cambie de color y se hace con la propiedad `Hover`, también se aplicara un color diferente cuando se le dé Clic al botón y esto se logra con la propiedad `Active`.

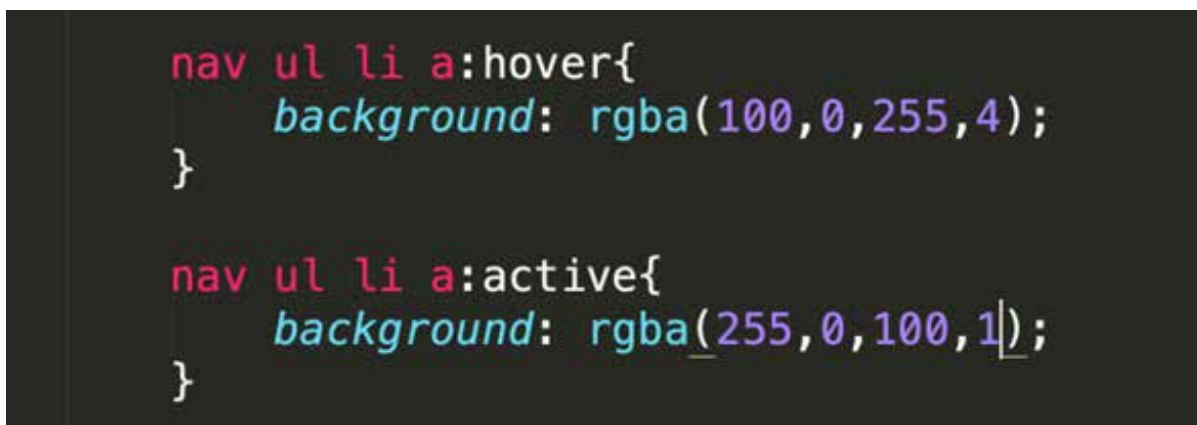


Ilustración 104. Propiedad Hover y Active para los botones.

Fuente: Aguilar, J. 2020

Después de configura toda la barra de navegación también llamada en este proyecto como sección 2 se seguirá adelante en el proceso.

La sección 3 se debe asignar 3 cajas, la primera tendrá el siguiente código HTML.

```
<section id="seccion3" class="fila">

  <div class="col-lg-4 col-md-4 col-sm-4 col-xs-12">
    <center>
      
      <h1>Lorem ipsum</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </center>
  </div>
</section>
```

Ilustración 105. Primer elemento de la sección 3.

Fuente: Aguilar, J. 2020.

Se debe recordar que las clases de columnas indican como se debe ver según el tipo de pantalla, si es una pantalla grande (lg) la caja debe ocupar 4 columnas, si es una pantalla mediana (md), la caja debe ocupar 4 columnas, si es una pantalla pequeña (sm) también debe ocupar 4 columnas, pero si es un dispositivo móvil (xs) la caja debe ocupar 12 columnas.



Ilustración 106. Logo

Fuente: Aguilar, J. 2022.

En el navegador se visualiza la página como muestra la imagen anterior. Ahora se le aplica el siguiente estilo CSS para darle un cambio en el diseño.

```
#seccion3 center{
  padding: 20px;
}

#seccion3 img{
  border-radius: 100%;
  width: 200px;
  height: 200px;
  padding: 10px;
}
```

Ilustración 107. Código CSS de las imágenes de la sección 3

Fuente: Aguilar, J. 2020

El código CSS mostrado en la imagen al aplicarse hace que la página y exactamente la sección 3 se muestre así.



Ilustración 108. Página visualizada desde navegador.
Fuente: Aguilar, J. 2020

Lo que falta es modificar el título de cada elemento, eso se hace con el siguiente código.



Ilustración 109. Estilo CSS etiqueta h1 sección 3
Fuente: Aguilar, J. 2020

Ahora lo que falta es replicar lo realizado anteriormente dos veces más para terminar la sección 3.

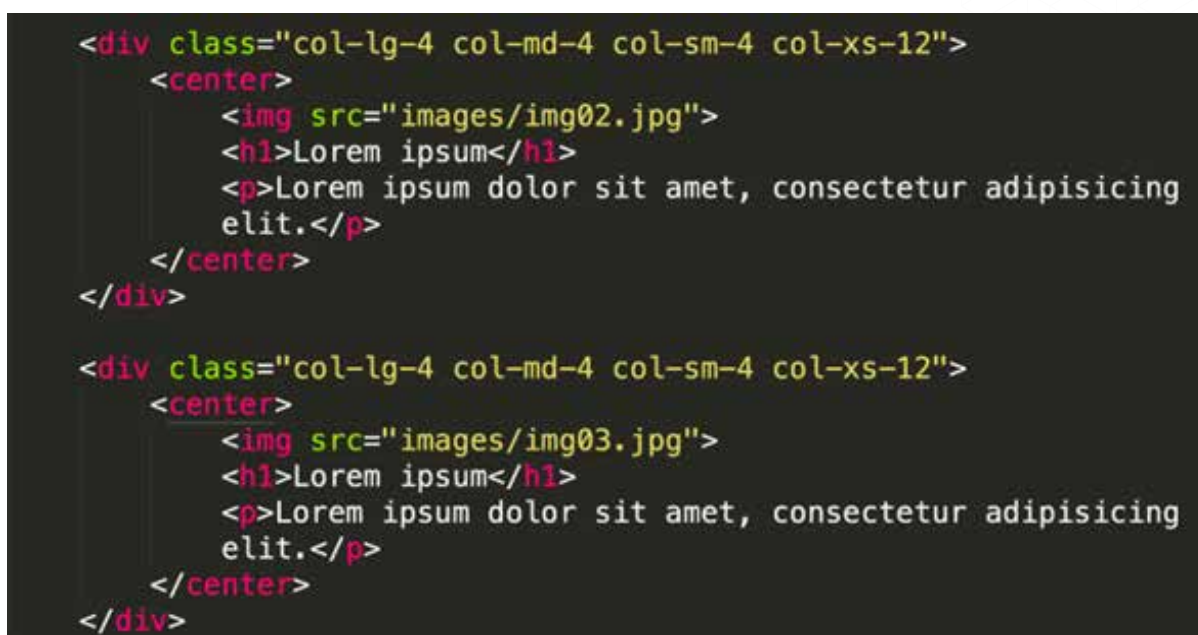


Ilustración 110. Agregando los otros dos div de la sección 3
Fuente: Aguilar, J. 2020

Terminando la sección 3 la página en diferentes pantallas se visualiza de la siguiente manera.

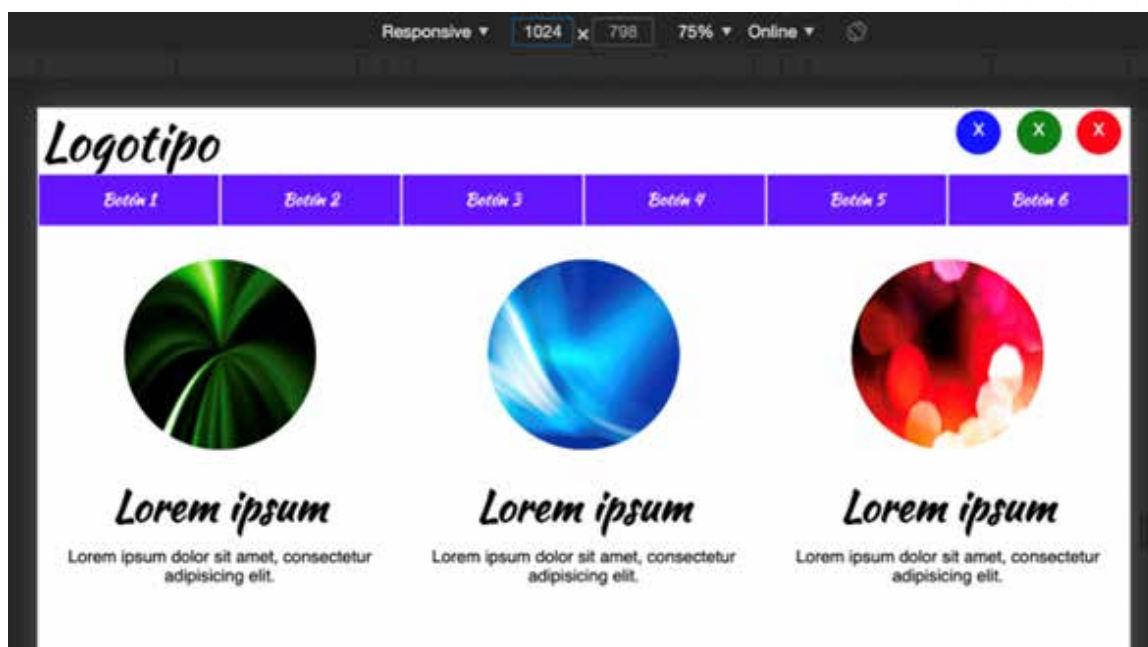


Ilustración 111. Visualización en 1024.
Fuente: Aguilar, J. 2020



Ilustración 112. Visualización en 800 pixeles.
Fuente: Aguilar, J. 2020



Ilustración 113. Visualización en 328 pixeles.
Fuente: Aguilar, J. 2020

Así como se puede observar en las diferentes pantallas, la página ya es responsiva, se visualiza organizada y de diferente forma de acuerdo al tipo de pantalla desde donde se está navegando. Para continuar se va a seguir con la sección 4, en este caso se va a crear un aside y un article.

```
<section id="seccion4" class="fila">
  <aside id="izq" class="col-lg-3 col-md-3 col-sm-3 col-xs-12">
    <ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
    </ul>
  </aside>
```

Ilustración 114. Creando elementos aside.
Fuente: Aguilar, J. 2020

Después de crear el aside, se pasa a crear el article.

```
<article class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
  <h1>Lorem Ipsum</h1>
  <p>Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint
occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.</p>
  <br>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit, sed do eiusmod tempor incididunt ut labore et dolore
magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat.</p>
</article>
```

Ilustración 115. Creando el article de la sección 4.

Fuente: Aguilar, J. 2020

Y se termina la sección 4 con el otro aside que contiene el formulario.

```
<aside id="der" class="col-lg-3 col-md-3 col-sm-3 col-xs-12">
  <h1>Ingreso</h1>

  <form>
    <input type="text" placeholder="Usuario">
    <input type="text" placeholder="Contraseña">
    <input type="submit" value="Enviar">
  </form>

</aside>
```

Ilustración 116. Aside del formulario.

Fuente: Aguilar, J. 2020

De acuerdo a lo anterior, la página se visualiza así:



Ilustración 117. Página con el código HTML.
Fuente: Aguilar, J. 2020

Ahora falta aplicar estilos.

```

/*****
SECTION 4
*****/

#seccion4 aside ul li{
    background: #aaa;
    padding: 10px;
    margin: 2px 10px;
    list-style: square;
    list-style-position: inside;
    list-style-image: url(../images/lista.png);
}

#seccion4 article h1{
    font-family: "Kaushan Script", cursive;
    font-size: 40px;
    margin: 20px;
}
    
```

Ilustración 118. Estilos sección 4 parte 1
Fuente: Aguilar, J. 2020

```
#seccion4 article p img{
  float: left;
  margin-right: 20px;
}

#seccion4 article p{
  margin: 5px 20px;
  font-size: 14px;
  text-align: justify;
}

#seccion4 aside#der h1{
  line-height: 50px;
  text-align: center;
  background: rgba(100,0,255,1);
  color: white;
  margin-bottom: 5px;
}
```

Ilustración 119. Estilos sección 4 parte 2.
Fuente: Aguilar, J. 2020

Lo anterior se visualiza de la siguiente forma en el navegador.



Ilustración 120. Visualización en pantalla grande
Fuente: Aguilar, J. 2020

Se puede ver que la página ya está prácticamente culminada y responsiva de acuerdo a toda la maqueta-ción realizada paso a paso, solo falta el footer.


```
<footer class="fila">

  <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">&copy;
  Todos los derechos reservados</div>

</footer>
```

Ilustración 121. Footer de la página.

Fuente: Aguilar, J. 2020

Como se puede observar el footer de la página ocupará siempre las 12 columnas en todos los tipos de pantalla. Ahora se le aplicarán los estilos pertinentes.

```
/******
SECCION 5
******/

footer{
  padding: 20px 10px;
}

footer div{
  background: rgba(100,0,255,1);
  text-align: center;
  line-height: 50px;
  color: white;
}
```

Ilustración 122. Código CSS para el footer.

Fuente: Aguilar, J. 2020

Lo anterior ya con la página terminada queda de la siguiente manera.



Ilustración 123. footer visualizado en el navegador.

Fuente: Aguilar, J. 2020

Con lo anterior se finalizó el proceso de maquetar la página web y además aplicar diseño responsivo. En la página w3layout se pueden encontrar ejemplos de páginas ya maquetadas y responsivas.