

2.6 Menú horizontal con estilos y formularios con estilos.

2.6.1 Menú horizontal

Gracias a los estilos CSS podemos convertir una lista `` en un menú horizontal, en una galería de fotos u en otros elementos.

Paso 1: Estructura HTML

Usaremos una lista desordenada `` con elementos de lista `` para crear los ítems del menú. Cada ítem del menú será un enlace `<a>` que llevará a diferentes páginas o secciones del sitio web.

```
<header>
<nav
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="#">Acerca de</a></li>
    <li><a href="#">Servicios</a></li>
    <li><a href="#">Portafolio</a></li>
    <li><a href="#">Contacto</a></li>
  </ul>
</nav>
</header>
```

Paso 2: Estilos CSS

Ahora que tenemos la estructura HTML, vamos a darle estilo al menú horizontal utilizando CSS. Queremos que el menú sea horizontal, quitar las viñetas y que los ítems estén alineados uno al lado del otro.

- Para quitar las viñetas, usaremos la propiedad `list-style-type:none`; en la etiqueta ``.
- Para alinear los ítems usaremos la propiedad `display:inline`; en las etiquetas `li`.

```
nav ul {list-style-type: none; /* Quitamos las viñetas de la lista */ }
nav li {display: inline; /* Ítems del menú en una línea horizontal */ }
```

Paso 3: Añadir colores y estilos personalizados al menú horizontal

Puedes personalizar los colores y estilos del menú para que se ajusten al diseño de tu sitio web. Por ejemplo, puedes cambiar el color de fondo del encabezado, los colores de los ítems del menú y los efectos de interacción al pasar el cursor sobre ellos.

De esta forma se consigue un menú horizontal completamente funcional y estilizado en tu sitio web. Puedes agregar más ítems o enlaces a medida que necesites, y con un poco más de CSS, puedes hacer que el menú sea aún más atractivo y fácil de usar para los visitantes.

Ejemplos de menú horizontal

A continuación se muestra un **ejemplo muy básico de menú horizontal que modifica su apariencia cuando se pasa el cursor por encima**.

HTML	CSS
<pre><header> <nav> Inicio Acerca de Servicios Portafolio Contacto </nav> </header></pre>	<pre>body { margin: 0; font-family: Arial, sans-serif; } header { background-color: #333; color: #fff; padding: 10px; } nav ul { list-style: none; padding: 0; margin: 0; } nav li { display: inline; /* Hace que los ítems del menú estén en una línea horizontal */ margin-right: 20px; /* Espacio entre los ítems del menú */ } nav a { text-decoration: none; color: #fff; font-size: 16px; } nav a:hover { text-decoration: underline; }</pre> 

En el siguiente ejemplo puedes ver una **barra de navegación con logotipo** bastante típica que puedes encontrar en cualquier web. Presta atención a que no se han utilizado clases, sólo se ha necesitado hacer uso de **selectores etiqueta** y **selectores descendientes**. Además fíjate que gracias al padding que se le ha incluido en los enlaces, serían perfectamente pulsables desde un dispositivo móvil.

HTML	CSS
<pre><header> <nav> </pre>	<pre>*{ margin: 0px; padding: 0px; font-family: "Roboto", Arial, Tahoma, sans-serif; } /* Header and Nav */ header{ position: fixed; width: 100%;</pre>

```

<li><a href="#">Inicio</a></li>
<li><a href="#">Servicios</a></li>
<li><a href="#">Contacto</a></li>
</ul>
</nav>
</header>

height: 70px;
background-color: #F7F9FA;
}
nav{
max-width: 1024px;
margin: 0 auto;
}
header img{
padding: 18px 10px;
}
nav ul{
float: right;
list-style: none;
text-align: center;
vertical-align: top;
margin: 0 10px;
}
nav li{
list-style-type: none;
display: inline;
}
nav li a{
font-size: 1em;
text-decoration: none;
line-height: 70px;
padding: 15px 10px;
color: black;
background-color: white;
border: 1px solid grey;
}
nav li a:hover{
color: white;
background-color: black;
}

```



En este ejemplo se puede apreciar un **menú desplegable o dropdown menu**. El menú contiene tres ítems principales: Inicio, Nuestros servicios y Contacto.

El ítem «Nuestros servicios» tiene un submenú desplegable que se oculta inicialmente y se muestra al pasar el cursor sobre él. El submenú está representado por otra lista desordenada (``) con un id «submenu». Dentro del submenú, hay tres elementos de lista (``) que contienen enlaces (`<a>`) con los nombres de los servicios ofrecidos.

HTML	CSS
<pre> <nav class="dropdownmenu"> Inicio Nuestros servicios <ul id="submenu"> </pre>	<pre> .dropdownmenu ul, .dropdownmenu li { margin: 0; padding: 0; } .dropdownmenu ul { background: gray; list-style: none; } </pre>

```

<li><a href="#">Marketing</a></li>
<li><a href="#">Social Media</a></li>
<li><a href="#">Web design</a></li>
</ul>
</li>
<li><a href="#">Contacto</a></li>
</ul>
</nav>

```

```

width: 100%; }
.dropdownmenu li {
float: left;
position: relative;
width: auto; }
.dropdownmenu a {
background: #30A6E6;
color: #FFFFFF;
display: block;
font: bold 12px/20px sans-serif;
padding: 10px 25px;
text-align: center;
text-decoration: none;
-webkit-transition: all .25s ease;
-moz-transition: all .25s ease;
-ms-transition: all .25s ease;
-o-transition: all .25s ease;
transition: all .25s ease; }
.dropdownmenu li:hover a {
background: #000000; }
#submenu {
left: 0;
opacity: 0;
position: absolute;
top: 35px;
visibility: hidden;
z-index: 1; }
li:hover ul#submenu {
opacity: 1;
top: 40px; /* adjust this as per top nav
padding top & bottom comes */
visibility: visible; }
#submenu li {
float: none;
width: 100%; }
#submenu a:hover {
background: #DF4B05; }
#submenu a {
background-color: #000000; }

```

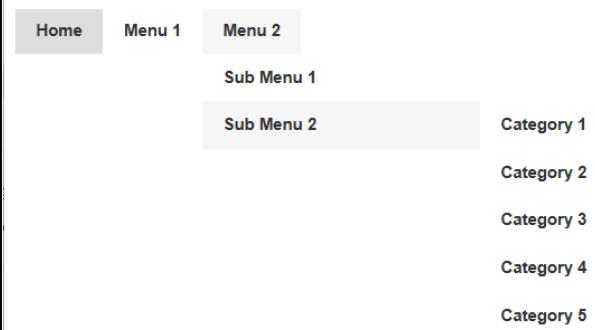


Por último, en este caso, el menú llega a tener hasta cuatro niveles de profundidad, lo que permite organizar y presentar una gran cantidad de opciones de manera jerárquica. Cada nivel de profundidad se representa mediante listas desordenadas anidadas dentro de otras listas, creando una estructura de árbol para la navegación del menú.

Recuerda siempre mantener un diseño limpio y sencillo para una mejor experiencia del usuario, así como de revisar el contraste entre el fondo y el texto. Además, asegúrate de dar un estilo diferente a los elementos pulsables cuando se les pasa el cursor por encima.

HTML	CSS
<pre> <h1> Menu con diferentes niveles</h1> <nav id="primary_nav_wrap"> <li class="current-menu-item">Home Menu 1 Sub Menu 1 Deep Menu 1 Sub Deep 1 Sub Deep 2 Sub Deep 3 Sub Deep 4 Deep Menu 2 Menu 2 <li class="dir">Sub Menu 1 <li class="dir">Sub Menu 2 Category 1 Category 2 Category 3 Category 4 Category 5 </nav> </pre>	<pre> #primary_nav_wrap{margin-top:15px } #primary_nav_wrap ul{ list-style:none; position:relative; float:left; margin:0; padding:0 } #primary_nav_wrap ul a{ display:block; color:#333; text-decoration:none; font-weight:700; font-size:12px; line-height:32px; padding:0 15px; font-family:"HelveticaNeue","Helvetica Neue",Helvetica,Arial,sans-serif } #primary_nav_wrap ul li{ position:relative; float:left; margin:0; padding:0 } #primary_nav_wrap ul li.current-menu-item{ background:#ddd } #primary_nav_wrap ul li:hover{ background:#f6f6f6 } #primary_nav_wrap ul ul{ display:none; position:absolute; top:100%; left:0; background:#fff; padding:0 } #primary_nav_wrap ul ul li{ float:none; width:200px } #primary_nav_wrap ul ul a{ line-height:120%; padding:10px 15px } #primary_nav_wrap ul ul ul{ top:0; left:100% } #primary_nav_wrap ul li:hover > ul{ display:block } </pre>

Menu con diferentes niveles



2.6.2 Formularios con estilos

Gracias a los estilos podemos romper el diseño gris con líneas negras de los formularios y convertirlos en una parte más que se integra perfectamente en nuestro diseño web.

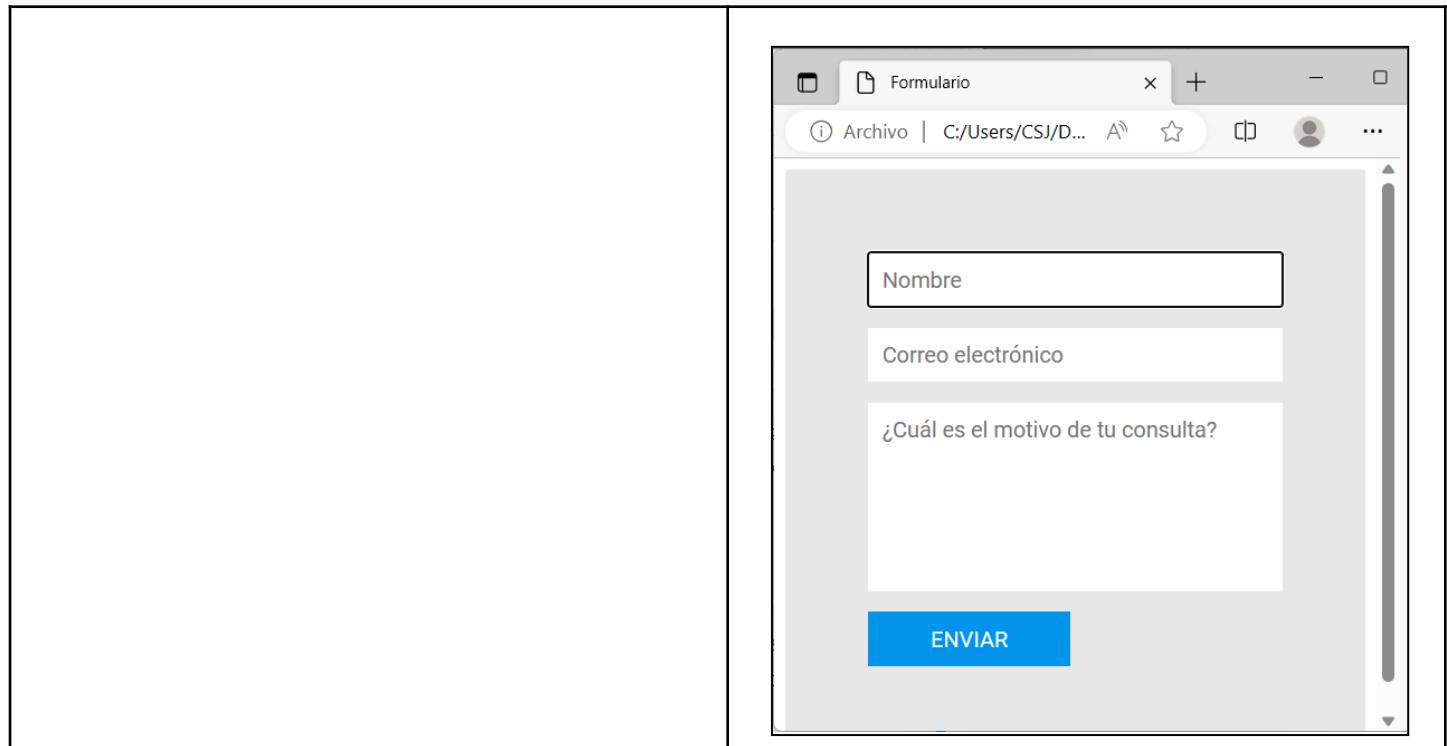
Como sabemos, a la hora de crear nuestros formularios, seguiremos buenas_prácticas_CSS, no crearemos clases extra innecesarias y utilizaremos, en la medida de lo posible, selectores descendientes y etiquetas predefinidas.

A las etiquetas de los formularios se les puede dar estilo CSS al igual que al resto de elementos de nuestro documento. Sin embargo, debemos conocer una serie de propiedades que nos ayuden a elaborar nuestros diseños de la forma más óptima posible.

Formulario básico

En el siguiente ejemplo puedes ver un formulario básico en CSS que permite ser visualizado en cualquier dispositivo:

HTML	CSS
<pre><form action="formulario-contacto.php" method="post"> <input name="nombre" type="text" placeholder="Nombre" maxlength="30" pattern="[a-zA-Z0-9]+" required autofocus/> <input name="email" type="email" placeholder="Correo electrónico" required /> <textarea name="consulta" placeholder="¿Cuál es el motivo de tu consulta?" rows="6" required></textarea> <button id="enviar" name="enviar" type="submit" class="btn">ENVIAR</button> </form></pre>	<pre>@import url(https://fonts.googleapis.com/css?family=Roboto:400); form{ padding: 60px; max-width: 400px; background-color: #E7E7E7; margin: 0 auto; } form input, form textarea{ margin-bottom: 15px; font-family: "Roboto", sans-serif; width: 100%; padding: 10px; box-sizing: border-box; border: none; color: #525c66; font-size: 1em; resize: none; } form button { display: block; background-color: #0095eb; padding: 10px 45px 10px 45px; border: 0; font-size: 1em; color: white; font-family: "Roboto", sans-serif; } form button:hover{ background-color: #046193; } textarea { resize: none; }</pre>



En este ejemplo se han utilizado muchas de las propiedades que hemos visto a lo largo del curso. Por ejemplo, se han quitado los bordes con la propiedad «border:none», se le ha dado un padding a los inputs y textarea, y se han utilizado las propiedades «margin: 0 auto» para centrar el contenido y «width: 100%» para conseguir un diseño web responsive.

Vamos a ver ahora ciertas propiedades que son de mucha utilidad en la creación de nuestros formularios.

Propiedad **box-sizing**

Por defecto en el modelo de cajas de CSS, el ancho y alto asignado a un elemento es aplicado solo al contenido de la caja del elemento. Si el elemento tiene algún borde (border) o relleno (padding), este es entonces añadido al ancho y alto del tamaño de la caja o contenedor. Esto significa que cuando se define el ancho y alto, se tiene que ajustar el valor para permitir cualquier borde o relleno que se pueda añadir.

La propiedad **box-sizing** puede ser usada para ajustar el siguiente comportamiento:

- **content-box** es el comportamiento CSS por defecto para el tamaño de la caja (box-sizing). Si se define el ancho de un elemento en 100 píxeles, la caja del contenido del elemento tendrá 100 píxeles de ancho, y el ancho de cualquier borde o relleno será añadido al ancho final desplegado.
- **border-box** tiene en cuenta cualquier valor que se especifique de borde o de relleno para el ancho o alto de un elemento. Es decir, si se define un elemento con un ancho de 100 píxeles. Esos 100 píxeles incluirán cualquier borde o relleno que se añada, y la caja de contenido se encogerá para absorber ese ancho extra. Esta propiedad es especialmente útil para

redimensionar cualquier elemento.

Vamos a ver un ejemplo: en el caso de los inputs, normalmente les añadimos un padding para darle algo de aire. Sin embargo, si después centramos el contenido y le dotamos de un «width:100%», este ancho no tendrá en cuenta el padding añadido al input y el elemento no se encontrará perfectamente redimensionado. Realiza la prueba quitando y añadiendo la propiedad «box-sizing: border-box» en el código del formulario básico creado en el punto 1.

Como puedes ver en el código, se han añadido los **prefijos para navegadores** necesarios para esta nueva propiedad.

```
-webkit-box-sizing: border-box;  
box-sizing: border-box;
```

Propiedad resize

Por defecto, los elementos <textarea> permiten cambiar el tamaño por el usuario. Podemos anular este comportamiento utilizando la propiedad «`resize:none`».

Valores: **none | both | horizontal | vertical**

Valores	Descripción
none	Permite cambiar el tamaño del elemento
horizontal	Permite cambiar el tamaño del elemento horizontalmente
vertical	Permite cambiar el tamaño del elemento verticalmente
both	Permite cambiar el tamaño del elemento en horizontal y vertical