

Práctica guiada FLEX

0. Objetivo

El objetivo de esta práctica es conseguir un header de una web con un aspecto similar al siguiente:

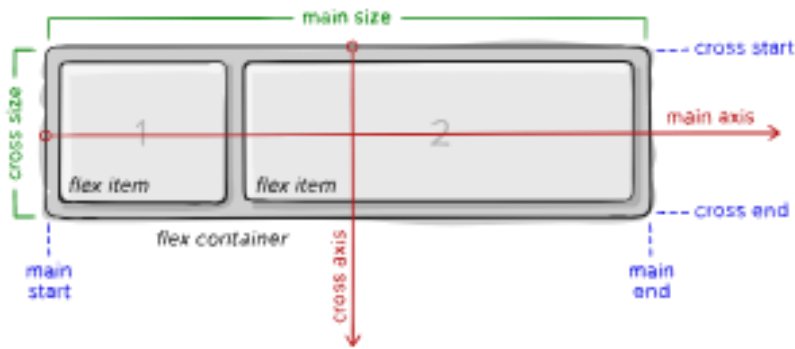


1. Flexbox

Ya es hora de ir empezando a usar características avanzadas de CSS. Vamos a empezar a usar el layout de flexbox.

El Flexbox Layoutmódulo (Caja Flexible) (una recomendación candidata del W3C a partir de octubre de 2017) tiene como objetivo proporcionar una forma más eficiente de diseñar, alinear y distribuir el espacio entre los elementos en un contenedor, incluso cuando su tamaño es desconocido y/o dinámico (de ahí el palabra “flexible”).

Dado que flexbox es un módulo completo y no una sola propiedad, implica muchas cosas, incluido su conjunto completo de propiedades. Algunos de ellos deben colocarse en el contenedor (elemento principal, conocido como "contenedor flexible") mientras que otros deben colocarse en los hijos (denominados "elementos flexibles").



Páginas de referencia usadas:

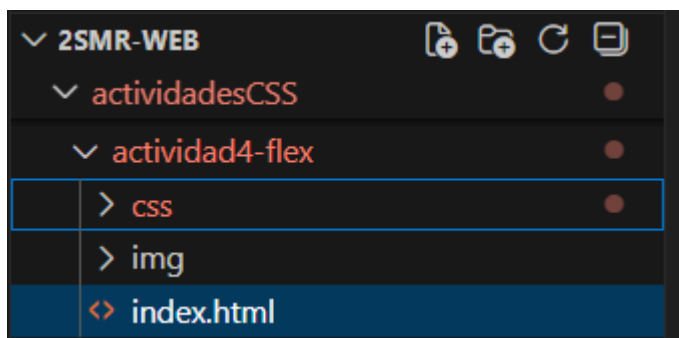
[A Complete Guide to Flexbox | CSS-Tricks](https://css-tricks.com/a-complete-guide-to-flexbox/)

https://www.w3schools.com/css/css3_flexbox.asp

2. Preparando nuestros iconos y logotipo

Lo primero que vamos a hacer es crear la estructura de directorios y ficheros de nuestro sitio web.

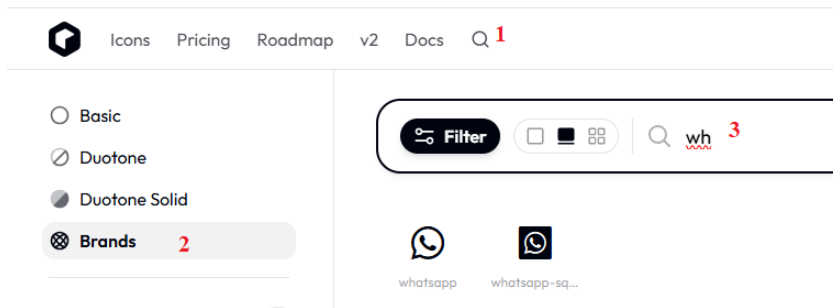
Vea esta imagen: Es la actividad4 que contiene la carpeta css y img así como el fichero index.html



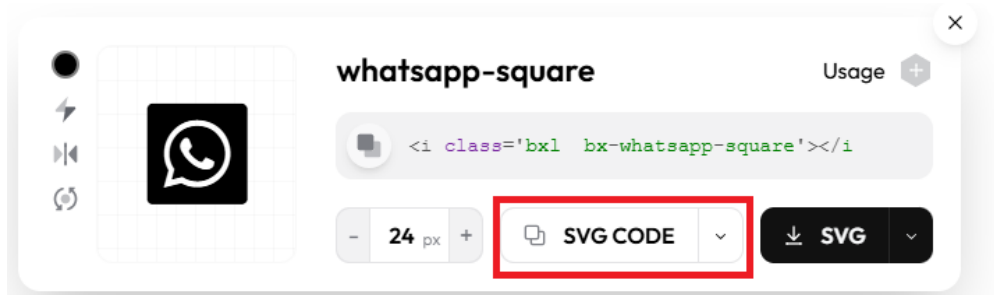
Descargar los iconos de las redes sociales.

De la web [Boxicons](https://boxicons.com/) vamos a descargar algunos iconos en formato svg, para usarlos en nuestro menú.

- Whatsapp
- Instagram
- Youtube

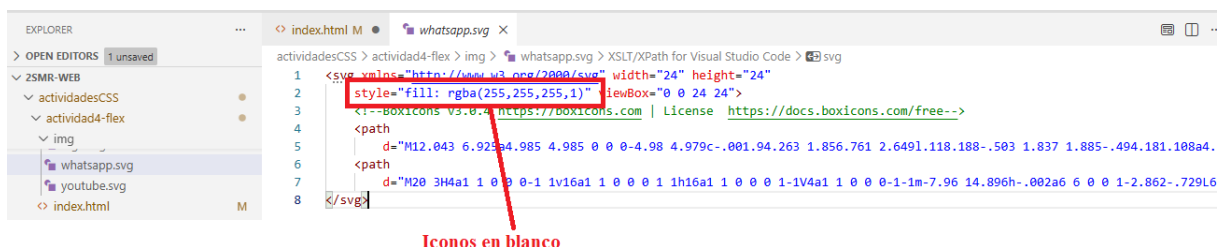


Selecciona la imagen encontrada y pulsa sobre **SVG CODE**



Tras haber copiado el contenido del icono en formato svg, nos iremos en Visual Studio Code (VSCode) a la carpeta **img**, crearemos un fichero lo llamaremos **whatsapp.svg** y pegaremos el código.

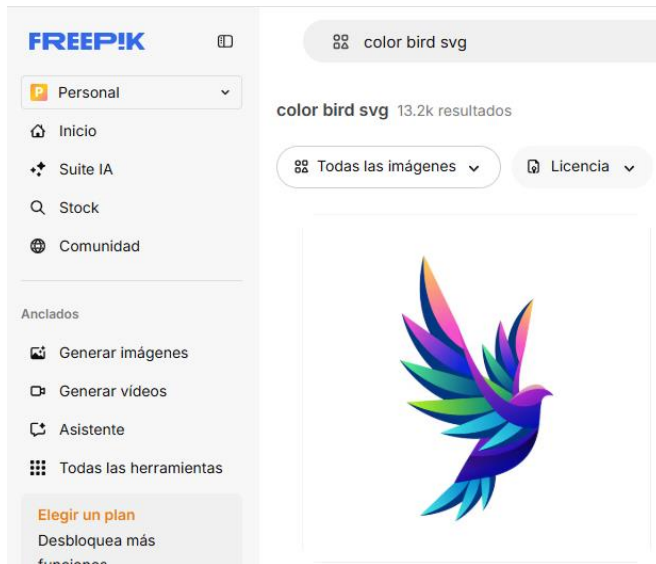
Este proceso lo vamos a repetir para el resto de iconos a usar



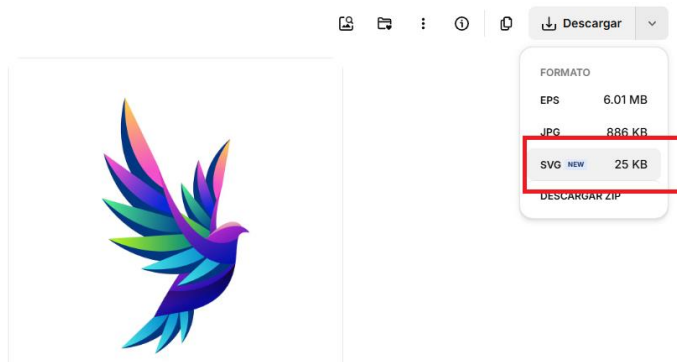
Este proceso lo vamos a repetir para el resto de iconos a usar (Instagram y YouTube)

*Cambia el valor de **fill** si quieres los iconos en blanco.*

Descargamos un logotipo para nuestra página web. Podemos usar esta web por ejemplo [Freepik](https://www.freepik.com)



Selecciona la imagen y descárgala en formato svg



Si queremos el fondo de la imagen transparente, ábrela en VSCode con el editor y pon el campo fill de rect al valor none



3. Creando nuestra estructura en html

Vamos a crear la siguiente estructura.

```

activadesCSS > actividad4-flex > <> index.html > html > body
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Actividad Flexbox</title>
7      <link rel="stylesheet" href="css/estilos.css">
8  </head>
9  <body>
10     <!-- Contenido aquí -->
11     <header class="mi_header">
12         <nav class="mi_menu mi_contenedor">
13             <div class="logo">
14                 <a href="#"></a>
15             </div>
16             <nav class="menu_nav">
17                 <a href="#">Inicio</a>
18                 <a href="#">Contacto</a>
19                 <a href="#">Acerca de</a>
20                 <a href="#">blog</a>
21             </nav>
22             <div class="menu_social">
23                 <a href="#"> </a>
24                 <a href="#"> </a>
25                 <a href="#"> </a>
26             </div>
27         </nav>
28         <section class="header_container mi_contenedor">
29             <h1 class="header_title">Vamos a aprender CSS</h1>
30             <p class="header_paragraph">
31                 Este es un sitio de prueba para practicar Flexbox en CSS.
32                 Cuando termines esta actividad, serás capaz de crear diseños flexibles
33                 y responsivos para tus proyectos web.
34             </p>
35             <a href="#" class="cta">Saber más</a>
36         </section>
37     </header>
38 </body>
39 </html>

```

La apariencia actual será la siguiente:



4. Aplicando estilos

En primer lugar vamos a crear una variable en la pseudoclase root para poder usarla desde todo el css.

En segundo lugar vamos a normalizar a 0 todos los margin y padding que usan los navegadores web y además indicamos que el tamaño de la caja va a ser a su borde.

Consulta esta web para obtener más información de la propiedad box-sizing [CSS box-sizing property](#)

```
1  /*vamos a crear una variable desde la pseudoclase root para poder acceder a
2  estas variable desde cualquier parte del documento */
3  :root{
4  |    --padding-container: 50px 0; /* arriba y abajo 100px, izquierda y derecha 0 */
5  |  }
6  /*reseteo de márgenes y paddings y box-sizing para que el padding y border se
7  incluyan dentro del width y height*/
8  *{
9  |    margin: 0;
10 |    padding: 0;
11 |    box-sizing: border-box;
12 |  }
13 }
```

Clase mi_contenedor

Le vamos a dar una anchura del 90% y una anchura máxima de 1200px.

Debemos centrar este contenedor y ocultar si algo sale de él.

Vamos a ponerle un fondo negro para ver que realmente se han aplicado los estilos correctamente. Este color de fondo lo eliminaremos después.

```
14 /* Es el contenedor de la página */
15 .mi_contenedor{
16 |    width: 90%;
17 |    max-width: 1200px;
18 |    margin: 0 auto; /* centramos verticalmente el contenedor */
19 |    overflow: hidden; /* para que no se salga nada del contenedor, si se sale se esconde*/
20 |    padding: var(--padding-container);
21 |    /* background-color: black;*/
22 | }
```

Vemos que ha cogido la anchura máxima de 1200px que aplica el padding de la variable --padding-container 50px arriba y abajo y que centra el contenedor con la propiedad margin: 0 auto.

Quitamos el color de fondo. (En el código aparece comentado)

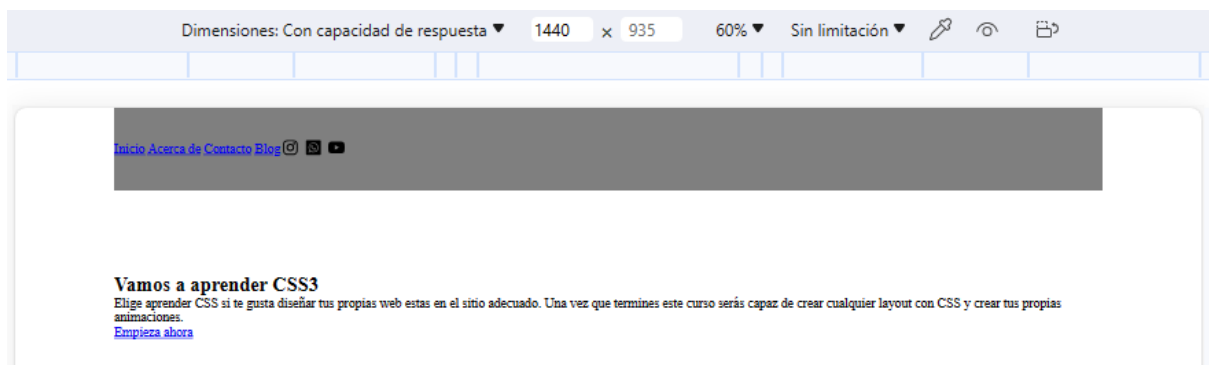
Clase mi_menu

Vamos a quitar el padding de la clase mi__container con la variable --padding-container. Aplicamos display flex, filas y nowrap y alineamos en el eje y a los items que hay dentro de mi__menu.

Damos una altura de 100px y ponemos un fondo momentáneo para ver la clase completamente, después lo eliminaremos.

La imagen del logo ha tomado 0px, vamos a intentar solucionar este inconveniente.

```
.mi_menu{
  /*background: rgba(0,0,0,0.5);*/
  display: flex; /* activamos flexbox */
  flex-flow: row nowrap; /* los elementos van en fila y no se envuelven */
  height: 100px;
  align-items: center; /* alineamos los elementos verticalmente al centro */
  --padding-container: 0; /* anulamos el padding del contenedor solo para el menu */
}
```



Le damos estilos a los ítems del contenedor mi_menu, consultar propiedades flex y realizar las pruebas que deseéis.

En un ítem de flex la propiedad **flex** es la forma corta de indicar flex-grow, flex-shrink y flex-basis

- **flex-grow** - Specifies how much a flex item will grow relative to the rest of the flex items
- **flex-shrink** - Specifies how much a flex item will shrink relative to the rest of the flex items
- **flex-basis** - Specifies the initial length of a flex item
- **flex** - Shorthand property for **flex-grow**, **flex-shrink**, and **flex-basis**

Los tres flex item que tiene el menú son: **el logo**, **los enlaces de menú** y **los iconos de redes sociales**.

```

.logo{
  flex: 0 1 auto; /* el logo ocupa su espacio natural */
  max-width: 90px; /* maximo ancho del logo */
  min-width: 60px;
  margin-left: 10px;
}

.menu_nav{
  flex: 2 1 70%; /* el menu ocupa 70% del espacio disponible */
  display: flex;
  justify-content: space-around; /* distribuimos los elementos del menu equitativamente */
}

.menu_nav a{
  text-decoration: none;
  color: white;
}

.menu_social{
  flex: 1 1 auto;
  display: flex;
  justify-content: space-around; /* distribuimos los elementos del menu social con espacio alrededor */
}

.menu_social a img{
  width: 32px;
  min-width: 28px;
}

```

Con estos estilos ya estaría posicionado el menú.



Clase mi_header

La clase mi_header es la clase que tiene todo el header, el resto de elementos están dentro de ésta.

Le vamos a dar el 100% del ancho, porque aquí colocaremos una imagen de fondo y queremos que ocupe todo el ancho de la pantalla, con respecto a la altura le vamos a dar 100vh y una mínima altura de 600px y una máxima altura de 800px (con esto en los ipads conseguimos que se vea correctamente).

Le vamos a dar una posición relativa para que después un elemento se posicione absolutamente a este contenedor.

```

.mi_header{
  width: 100%;
  height: 100vh; /* altura de la ventana del navegador 100vh = full height of the viewport*/
  min-height: 600px;
  max-height: 800px;
  position: relative;
}

```


Clase mi_header::before

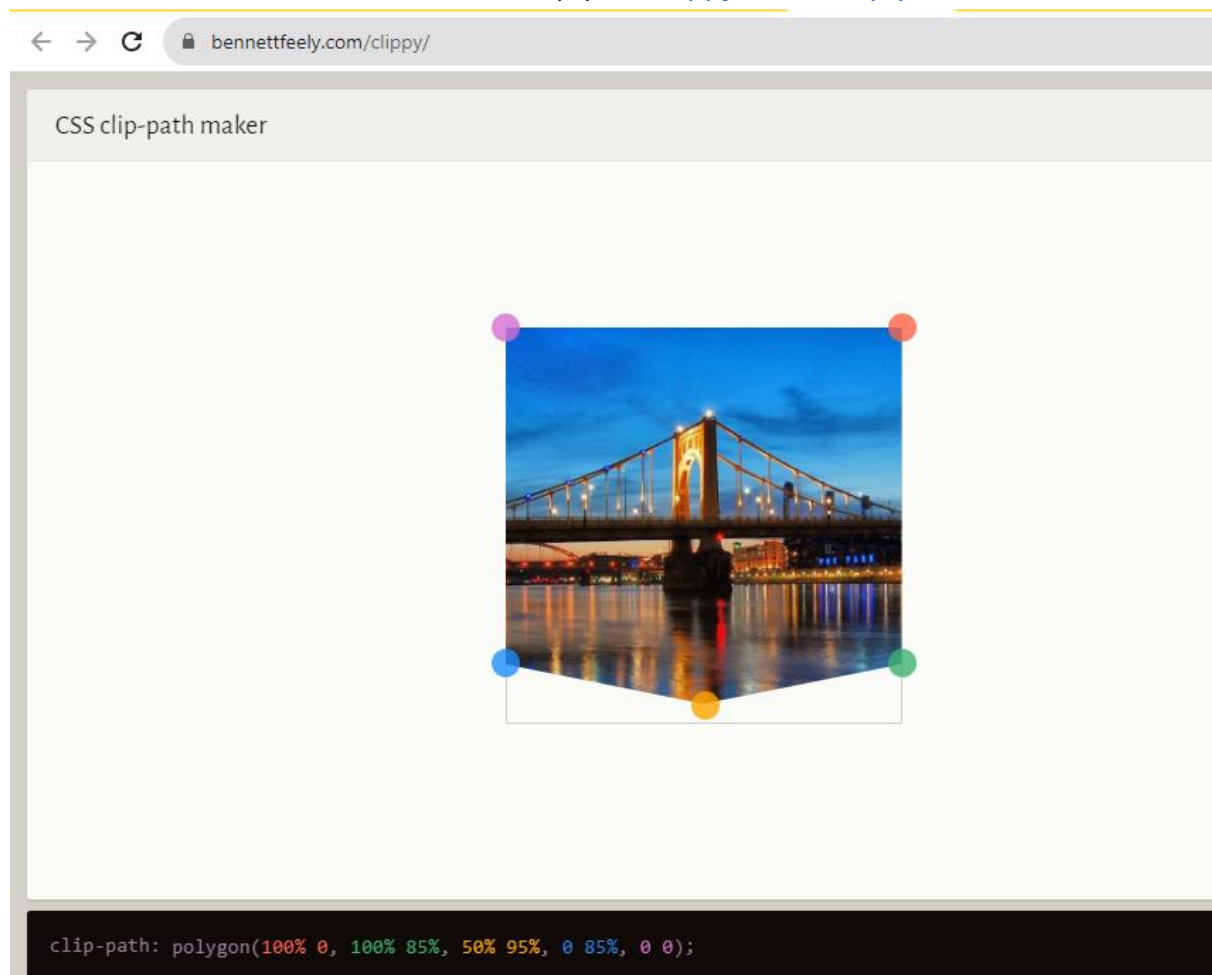
Queremos poner una imagen de fondo y aplicar un [clip-path](#) (**clip-path nos va a permitir dar formas a las imágenes**).

Es decir, todo el header va a tener como fondo una imagen.

Hay un problema si queremos usar la posición fixed y clip-path. Si la posición fixed es hijo de clip-path automáticamente en vez de fijarse al navegador se fija al elemento clip-path.

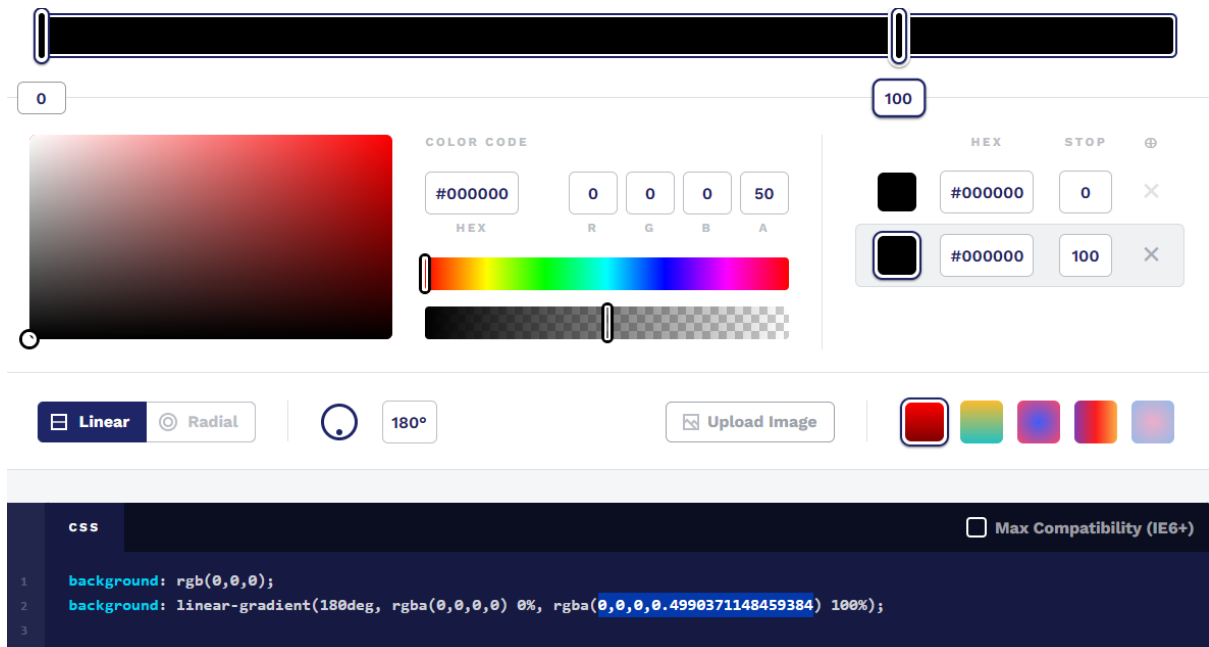
Debido a esto, vamos a crear un pseudo elemento a partir de mi_header, el pseudo elemento ::before [CSS ::before Selector](#)

Antes de nada vamos a crear nuestro clip-path [Clippy — CSS clip-path maker](#)



Copiamos el código.

También vamos a usar un gradiente sobre la imagen [CSS Gradient — Generator, Maker, and Background](#)



Nos quedaremos con la línea **linear-gradient**

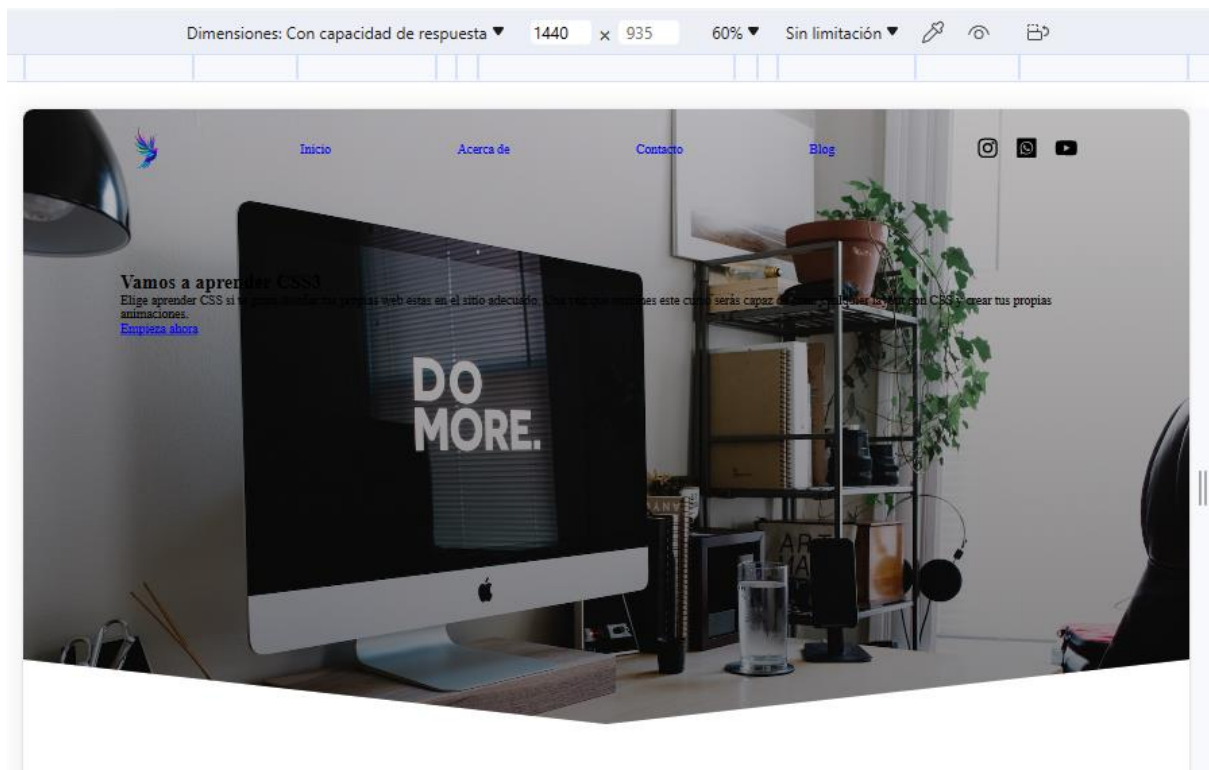
Estas propiedades hacen que la imagen de fondo se vea mejor, es cuestión de probar cómo queda para obtener el diseño que se ajuste a nuestra página.

Las imágenes las podemos buscar en pixabay y en unplash

```

.mi_header::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: linear-gradient(
    180deg,
    rgba(0, 0, 0, 0.25) 0%,
    rgba(0, 0, 0, 0.5) 100%
  ),
  url("../images/fondo_header.jpg");
  background-size: cover;
  clip-path: polygon(100% 0, 100% 85%, 50% 95%, 0 85%, 0 0);
  z-index: -1;
}

```



Coloreando el menú

Como ya se ha comentado anteriormente estas propiedades pondrán el menú y las redes sociales en color blanco.

```

.v mi__nav a {
  text-decoration: none;
  color: #FFFFFF;
}

```



Hacemos este cambio en todos los iconos de red social



Vamos a por el texto del header.

Clase header_container

```
.header_container{
  height: 75%;
  display: flex;
  flex-direction: column; /* los elementos van en columna */
  justify-content: center; /* centramos verticalmente */
  color: white;
  text-align: center;
  max-width: 800px;
}

.header_title{
  font-size: 3rem;
  padding: 20px 0;
}

.header_paragraph{
  font-size: 1.2rem;
  line-height: 2rem;
  margin-bottom: 25px;
}

.cta{
  display: inline-block;
  background-color: blue;
  align-self: center; /* centramos el boton horizontalmente */
  color: white;
  padding: 13px 30px;
  border-radius: 32px;
}
```



¿Cómo pondrías el logo en blanco?

5. Reto

Consigue esta apariencia.

