

MANUAL INKSCAPE GIMP AUDACITY SVG/CANVAS HECHO POR JESÚS ENRIQUE ROZAS PENA



**El objetivo final de este tutorial es realizar lo siguiente,
Cambiar la tipografía y los colores del logo de HTML5**

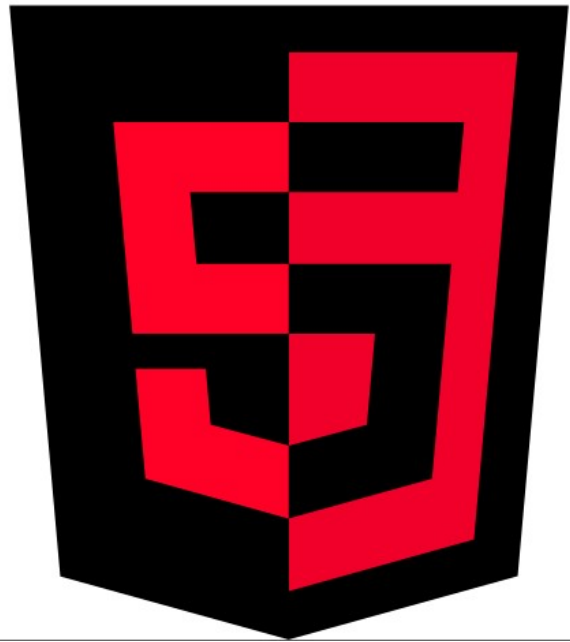
ANTES:

HTML



DESPUÉS:

WEB



Lo primero que tenemos que hacer es elegir la imagen, que al trabajar con inkscape habrá que utilizar una imagen **.svg**

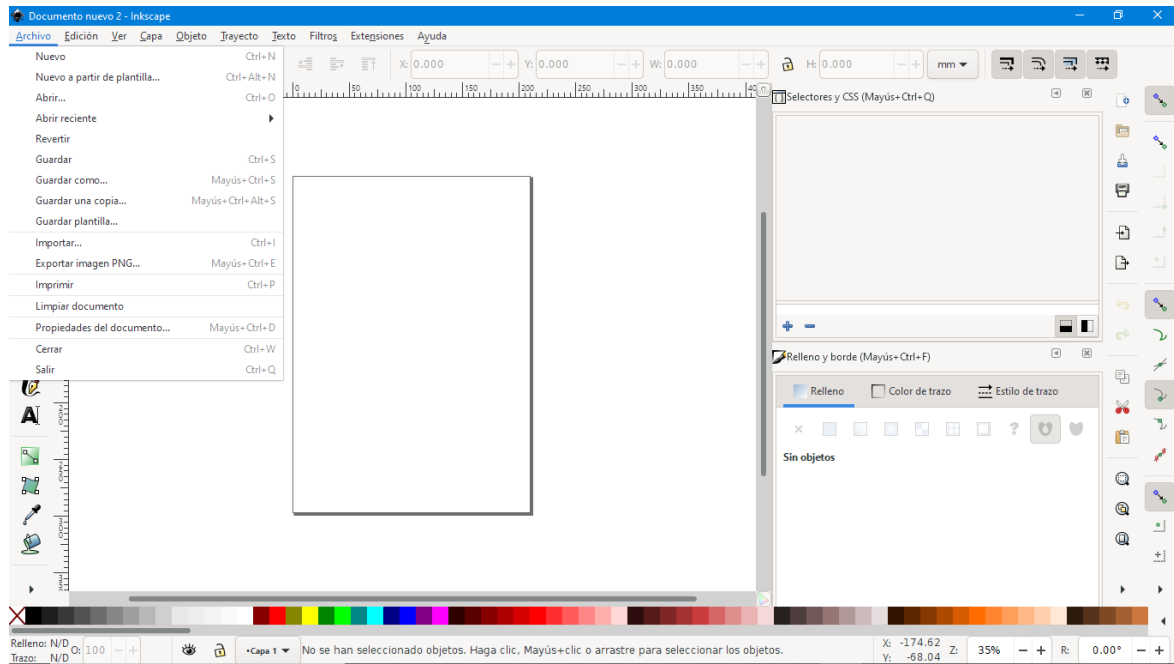
En el siguiente link podréis conseguir la imagen que yo utilice para realizar este ejercicio
https://commons.wikimedia.org/wiki/File:HTML5_logo_and_wordmark.svg

1. Modifica utilizando Inkscape el logo de HTML para que los colores sean rojo y negro y cambia el texto "HTML5," por "WEB"

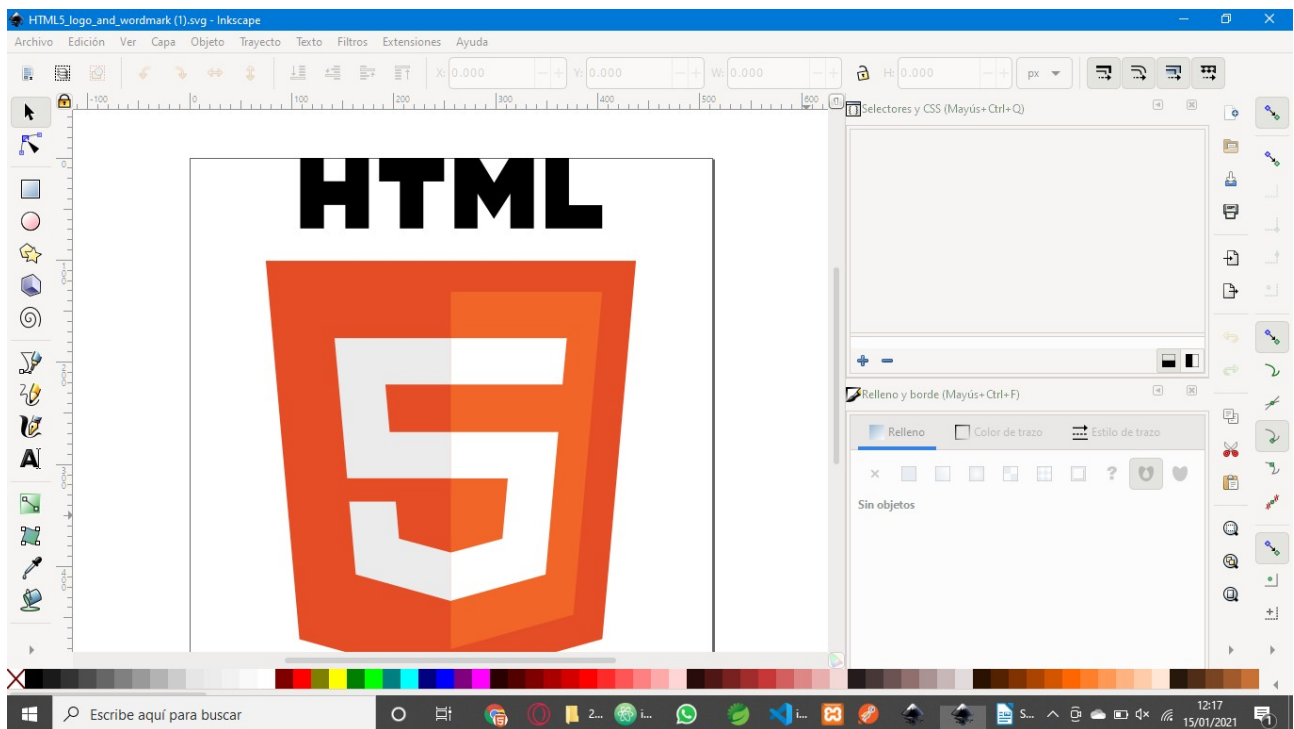
1. Comienzo

(Antes de todo, tendremos que instalar inkscape <https://inkscape.org/release/inkscape-1.0.1/>).
Lo primero sera abrir el archivo que acabamos de descargar

Archivo > Abrir > BUSCAREMOS EN LA CARPETA QUE GUARDAMOS EL ARCHIVO > LO SELECCIONAREMOS



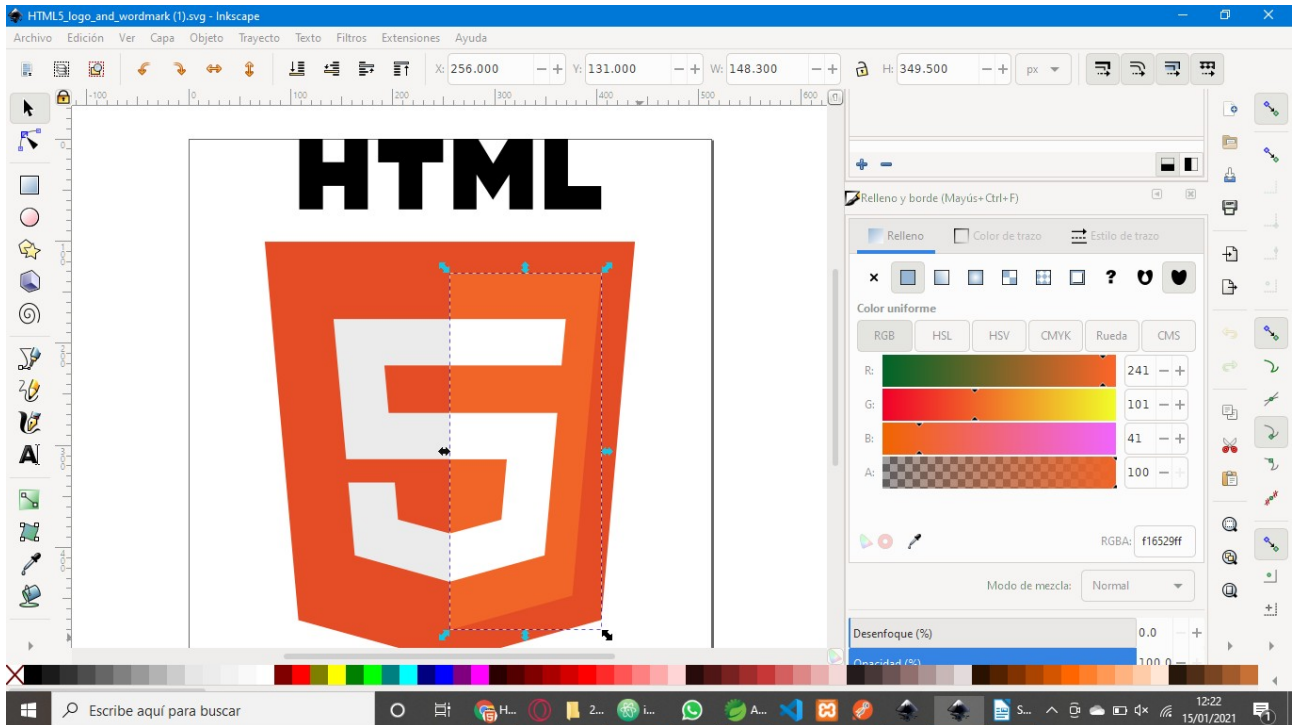
realizar todo este proceso, sencillo, tendremos que visualizar lo siguiente, que confirmaría que lo hicimos todo correctamente:



(Si para hacer este ejercicio usamos una imagen que no sea .svg, no nos permitirá seleccionar cada parte de la imagen, haciendo imposible realizarlo)

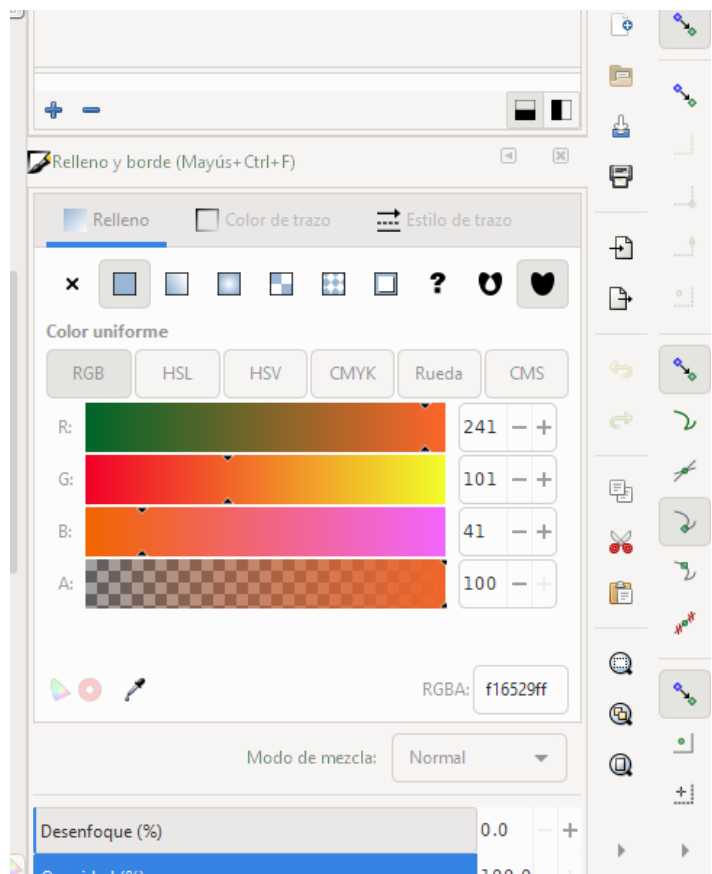
1.1.1 - Cambio de color

Ahora seccionaremos cada parte que queramos cambiar de color, **una a una**.



Al seleccionar cada parte cambiaremos de color, usando los indicadores que os muestro a la derecha

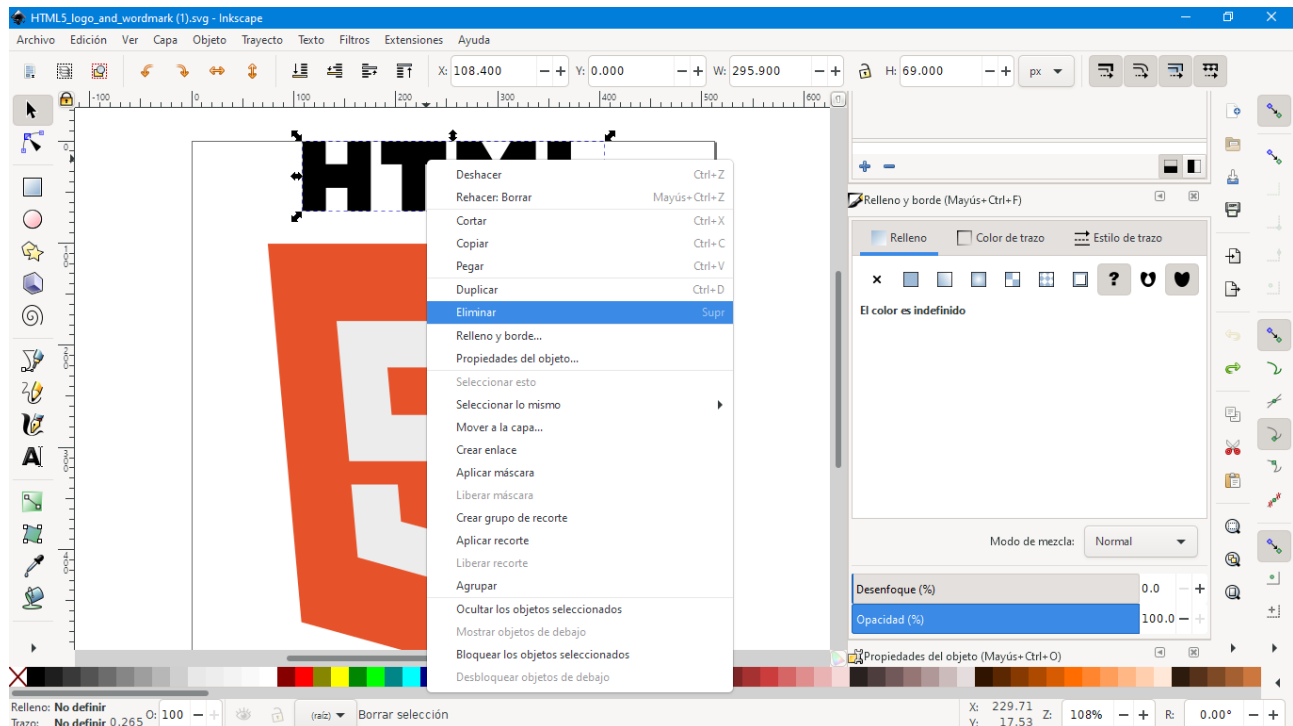
Podremos cambiarlo a cualquier color que queramos. Aunque para este ejercicio tendremos que cambiarlo a rojo y negro obligatoriamente.



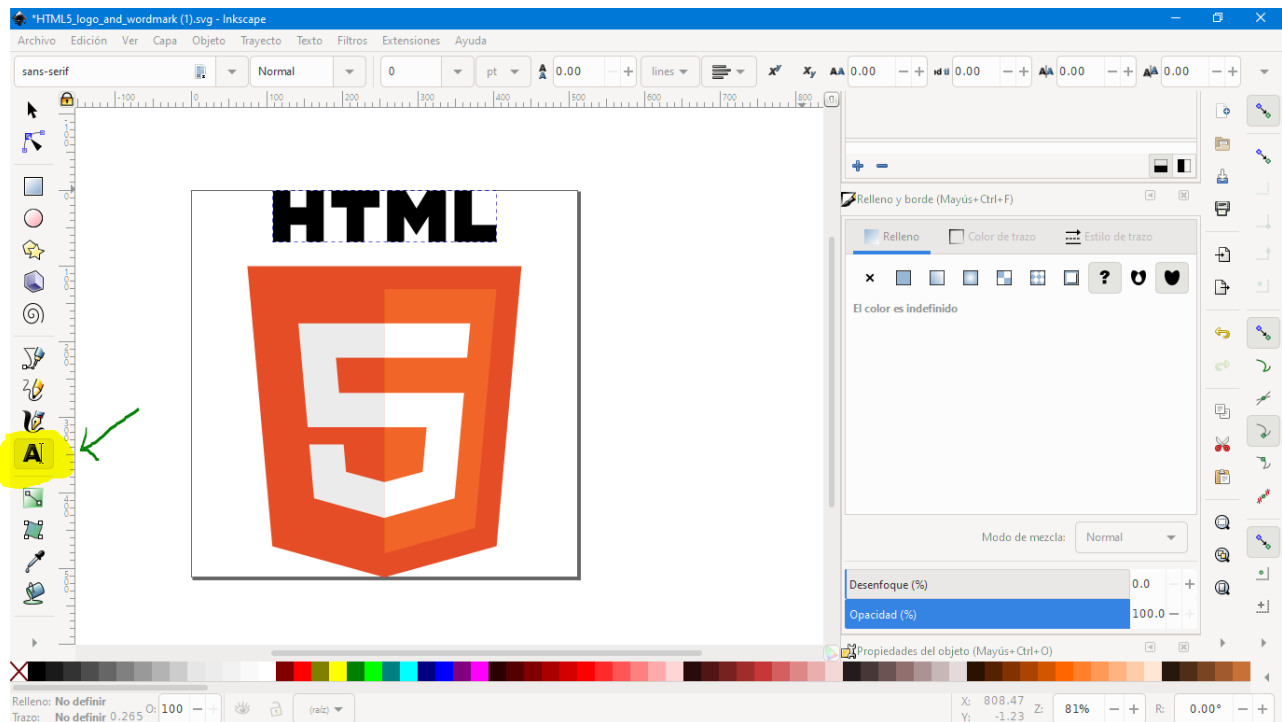
1.1.2 Tipografía

Para acabar el ejercicio, tendremos que cambiar el texto de **HTML** a **WEB**.

Para realizar esto seleccionaremos el texto HTML y lo borrarémos/eliminaremos.

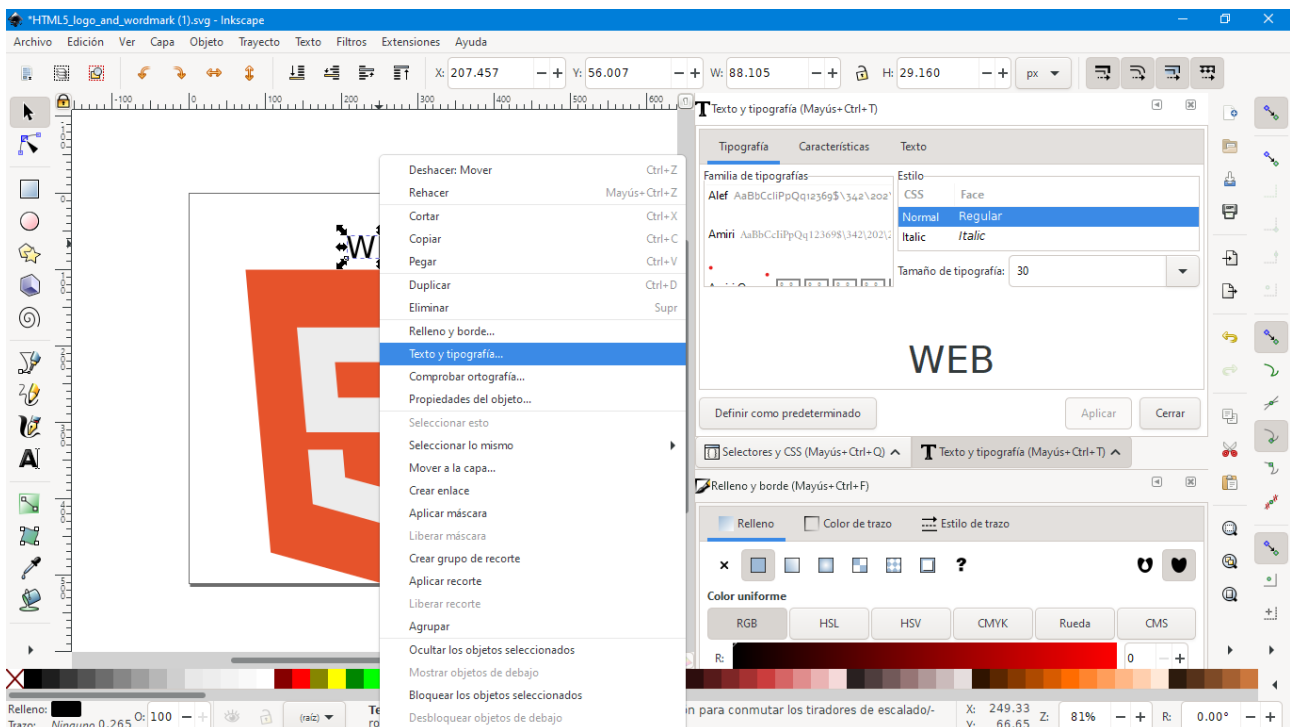


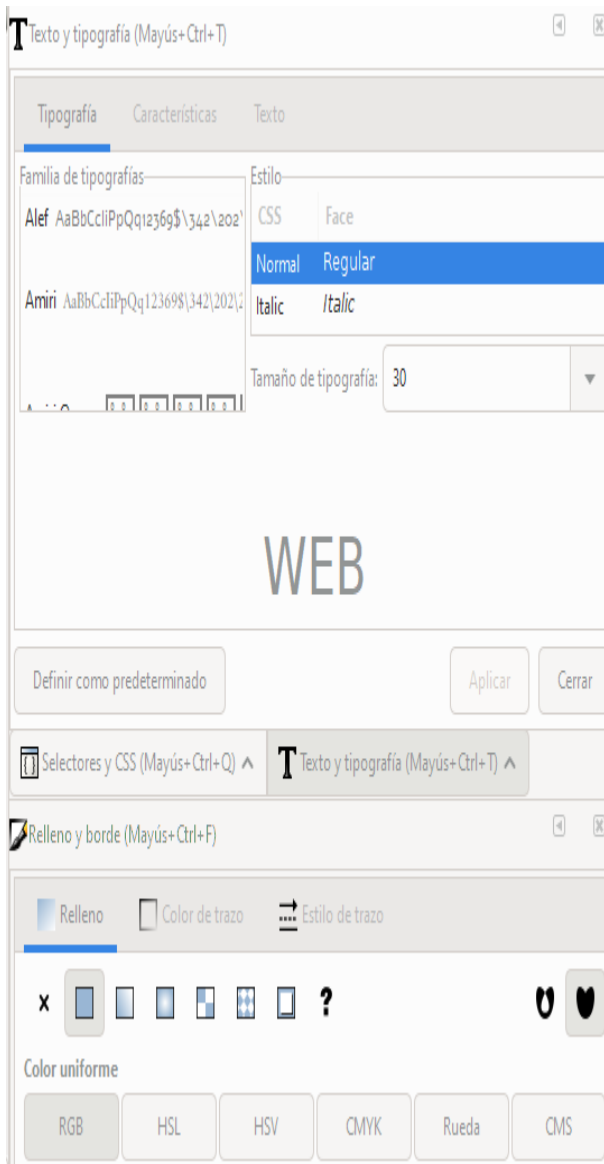
A continuación para crear un nuevo texto iremos al siguiente apartado:



El siguiente paso sería escribir lo que queramos tener en lugar de HTML, en nuestro caso obligatoriamente será WEB.

Para cambiar los estilos de nuestro texto tendremos que hacer lo siguiente
→ click derecho > texto y tipografía





Gracias a este apartado podremos modificar los estilos de nuestro texto.

En nuestro caso se tendrá que aparecer lo máximo posible al formato usado en el de la imagen Original

1. 2 - Comienzo

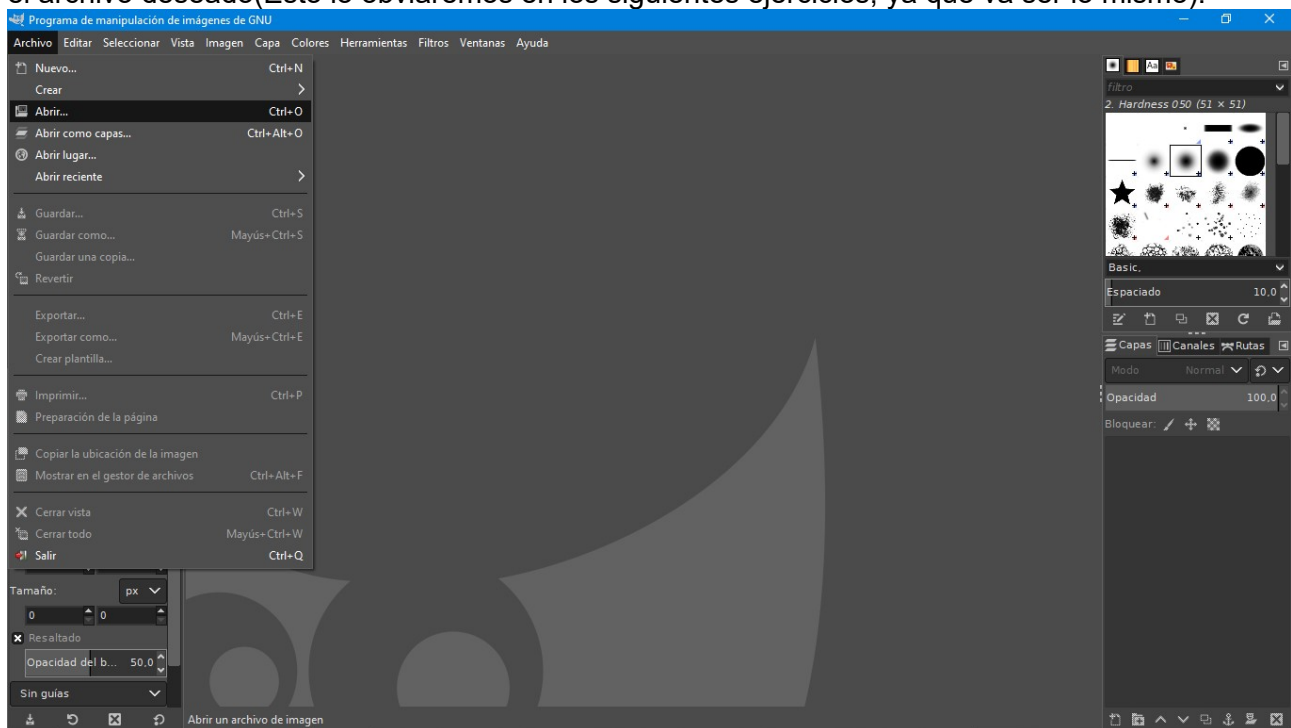
En el siguiente ejercicio tendremos que usar GIMP (lo primero como en el anterior sería descargarlo desde esta URL: <http://www.gimp.org.es/descargar-gimp.html>)

El ejercicio consistirá en, usando la imagen que editamos anteriormente, cambiar todos los formatos de la imagen. La cambiaremos a .GIF, .JPG y .PNG.

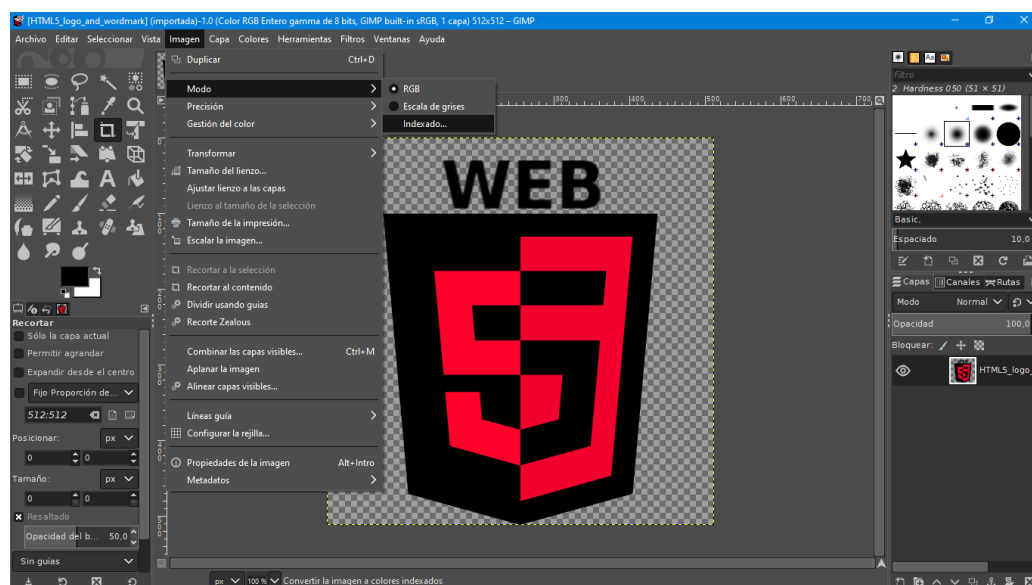
1.2.1 – GIF

Lo primero que haremos será pasar la imagen SVG a GIF.

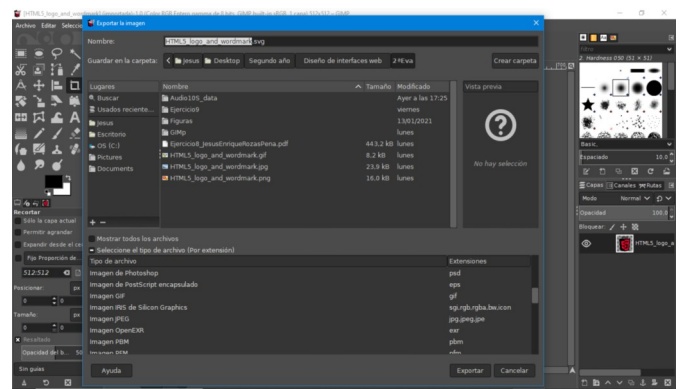
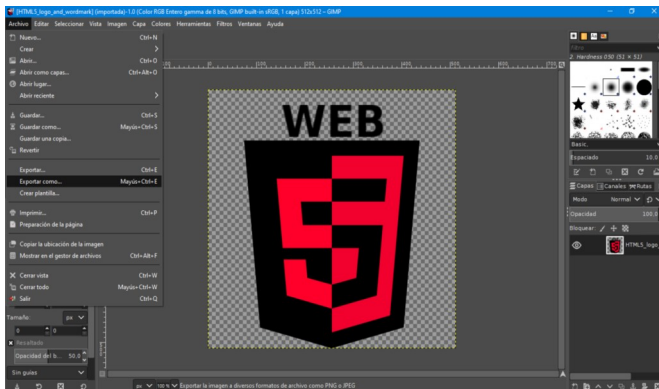
El primer paso tendremos que abrir el archivo .svg, de esta manera Archivo > Abrir > Seleccionar el archivo deseado (Esto lo obviaremos en los siguientes ejercicios, ya que va a ser lo mismo).



El ejercicio nos pide que los colores sean indexados, esta opción es muy sencilla de realizar, y se hará de la siguiente manera: imagen > modo > indexado

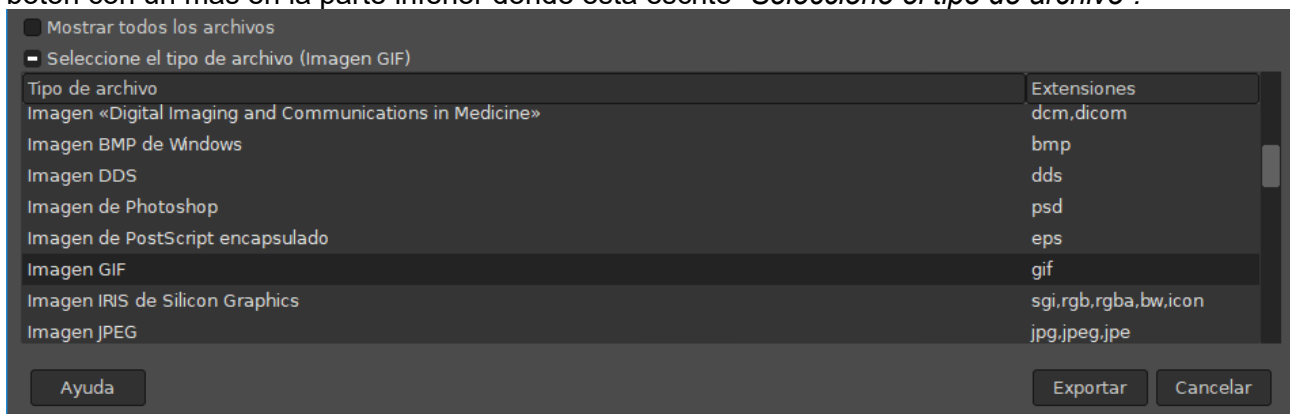


Para cambiar la imagen en los diferentes formatos tendremos exportar como, que estará en Archivo > Exportar como > Nos abrirá una ventana con la que trabajaremos (Este paso lo obviaremos en las dos siguientes, ya que siempre sera igual)

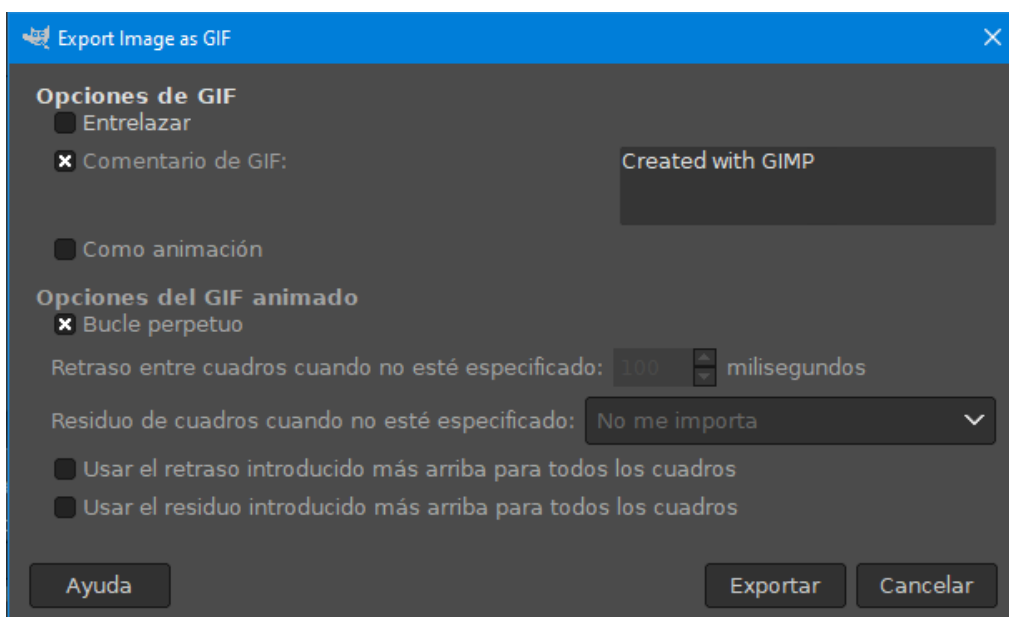


Estaremos entrando en esta ventana en todo los formatos

Una vez dentro de la ventana en la que entramos en el paso anterior, tendremos que darle al boton con un mas en la parte inferior donde esta escrito *“Seleccione el tipo de archivo”*.



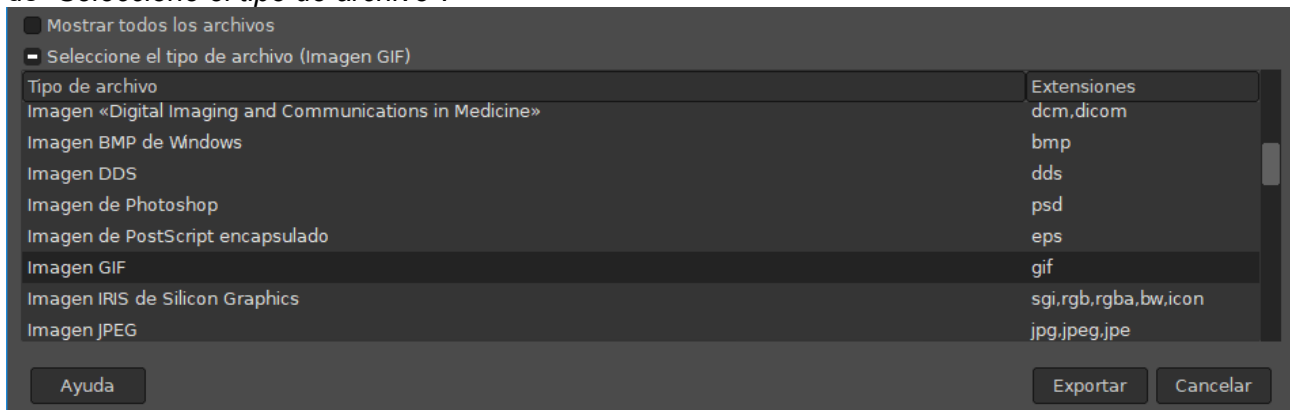
En este “pequeño” submenu, tendremos que buscar los formatos de las imágenes que queramos. Que en este caso sera GIF, por ultimo habrá que pulsar en el botón de Exportar.



Nos saldrá esta ventana para confirmar que queremos convertirlo en un gif. Y ya estaría creado.

1.2.1 – PNG

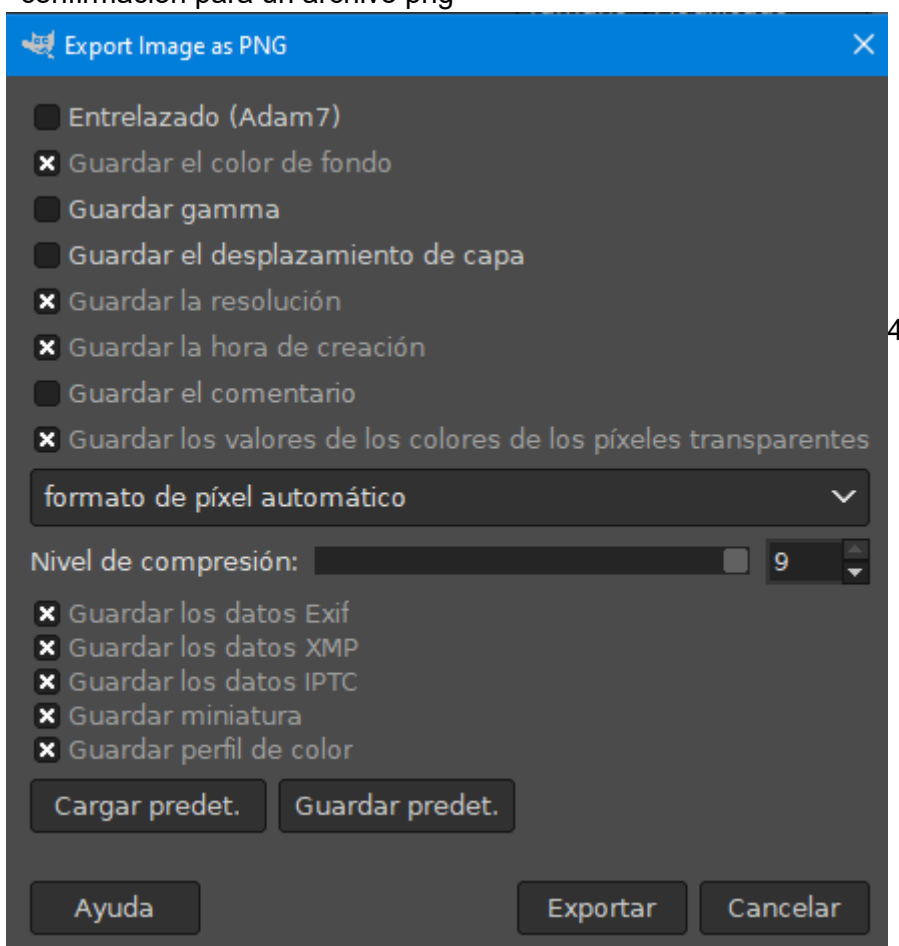
Los primeros pasos para convertir una imagen en .png serán los mismos, ata llegar a la ventana de “*Seleccione el tipo de archivo*”.



En esta ventana, al contrario que en el caso anterior, tendremos que seleccionar el formato .PNG



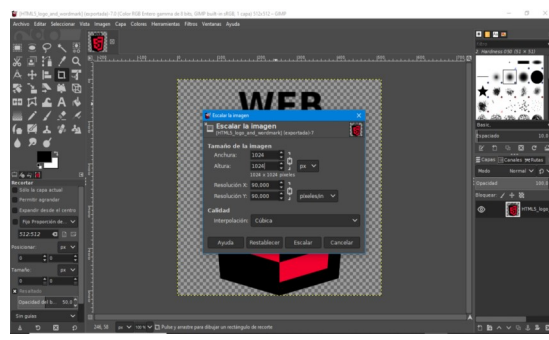
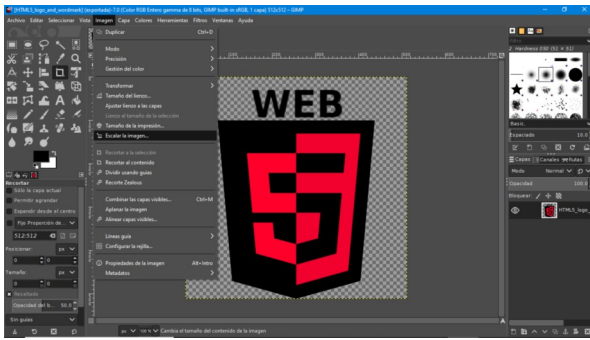
En vez de salir la ventana de confirmación de GIF, nos saldrá otra ventana diferente de confirmación para un archivo png



En este ejercicio tendremos que cambiar las dimensiones de la imagen, para realizar dicho este apartado tendremos que entrar en

10

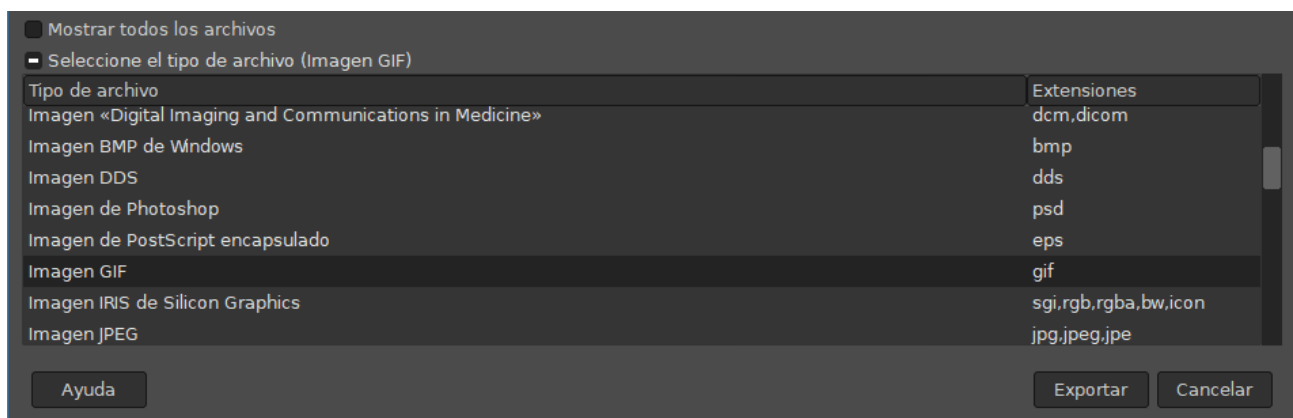
imagen > escalar Imagen > Y nos saldrá una ventana donde podremos cambiar las dimensiones de la imagen



En este ejercicio tendremos que cambiar la altura al doble que tenía la imagen original

1.2.2 – JPG

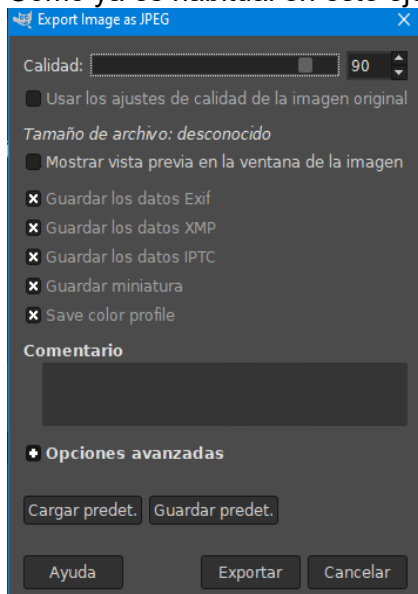
Los primeros pasos para convertir una imagen en .png serán los mismos, ata llegar a la ventana de “*Seleccione el tipo de archivo*”.



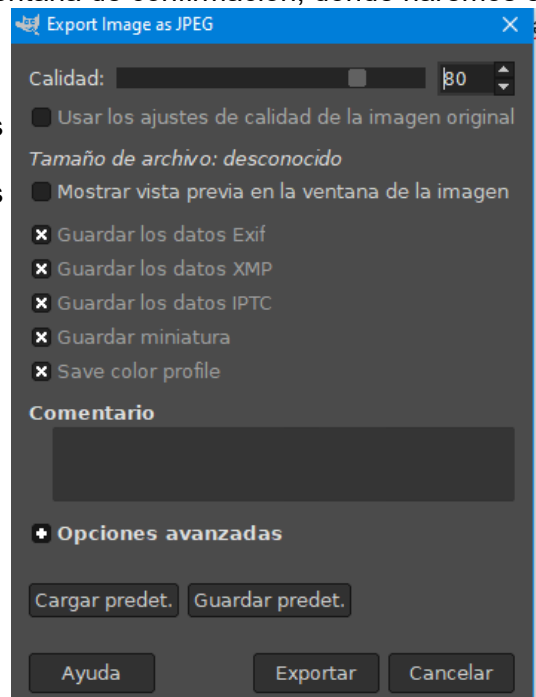
A diferencia de los dos anteriores, seleccionaremos en este caso jpeg



Como ya es habitual en este ejercicio, nos saldrá una ventana de confirmación, donde haremos el



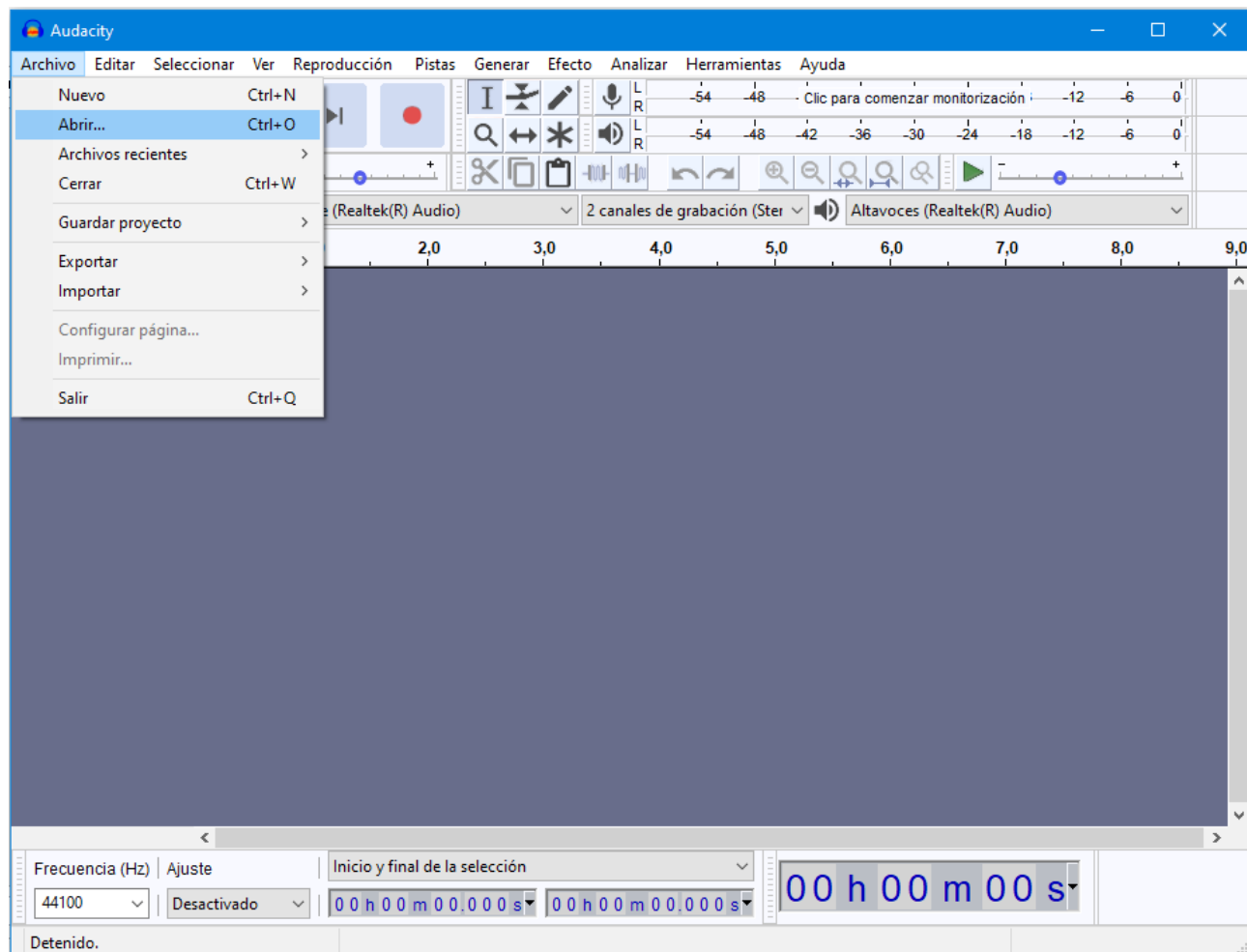
siguiente ejercicio que se trata de bajar la calidad. La imagen, por defecto nos saldrá con una calidad del 90 % y la tendremos que bajar al 80%



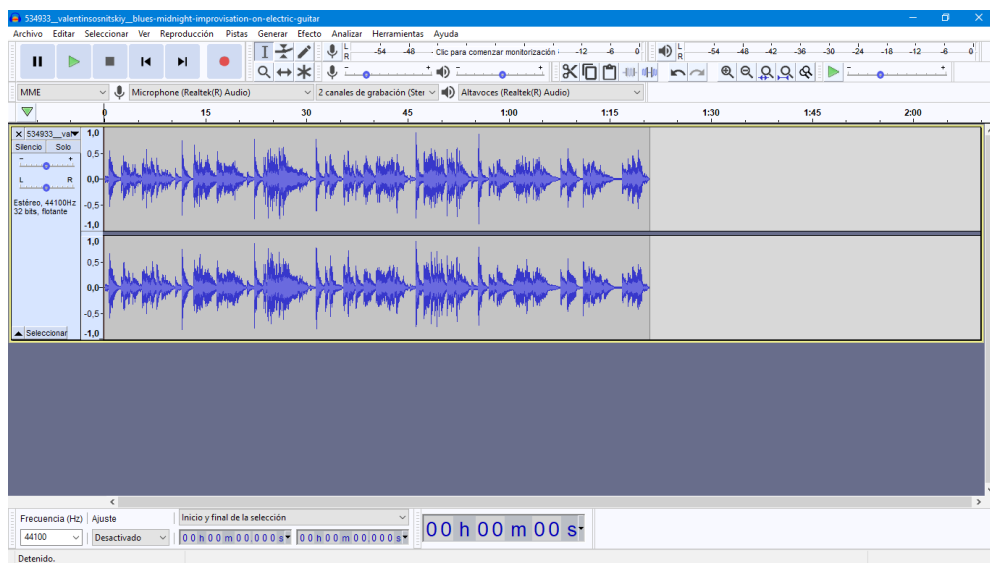
1.3 Audacity

Como en los dos ejercicios anteriores tendremos que descargar un programa para poder realizarlo

Para ello utilizaremos el siguiente link <https://audacity.es/>

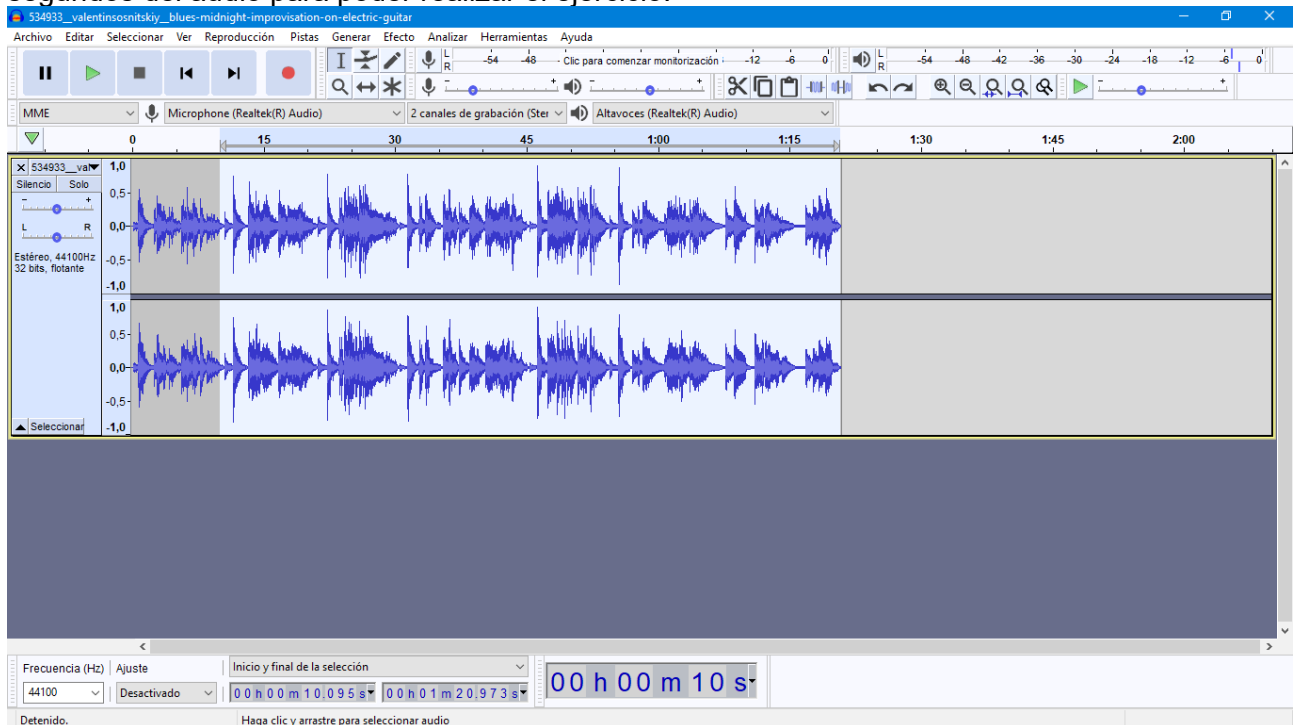


Abrimos el archivo deseado, Siguiendo la captura anterior, dando
Archivo > Abrir > Y buscar el archivo deseado para editarlo

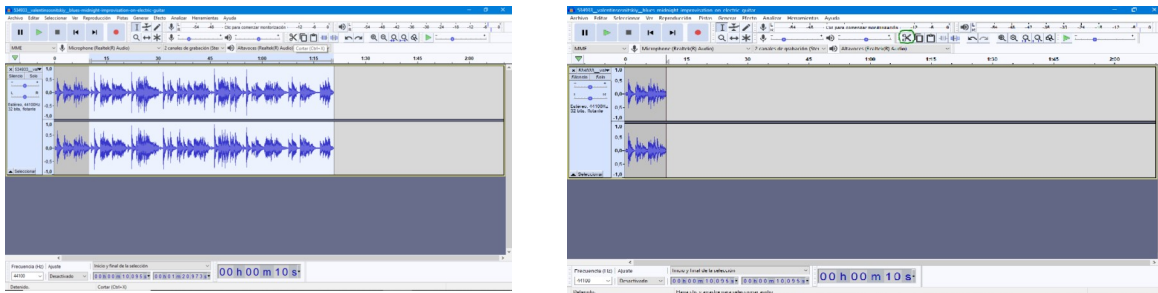


Una vez abierto, se
vería como en la
siguiente captura

Seleccionamos la parte del audio, que deseamos, que en este caso serán los primeros 10 segundos del audio para poder realizar el ejercicio.



Como ultimo paso vamos eliminar la parte seleccionada, para que nuestro audio solo dure 10 segundos.



Para cortar la imagen lo único que tendremos que hacer es pulsar el botón con forma de tijera que hay arriba a la derecha.

1.4 -SVG / CANVAS

Antes de realizar el ejercicio tendremos que tener en cuenta que es cada elemento:

- **SVG** → Gráficos vectoriales escalables, o gráficos vectoriales redimensionables o SVG es un formato de gráficos vectoriales bidimensionales, tanto estáticos como animados, en formato de lenguaje de marcado
- **CANVAS** → Canva es un software y sitio web de herramientas de diseño gráfico simplificado, fundado en 2012. Utiliza un formato de arrastrar y soltar y proporciona acceso a más de 60 millones de fotografías y 5 millones de vectores, gráficos y fuentes.

En este ejercicio nos pide varios elementos:

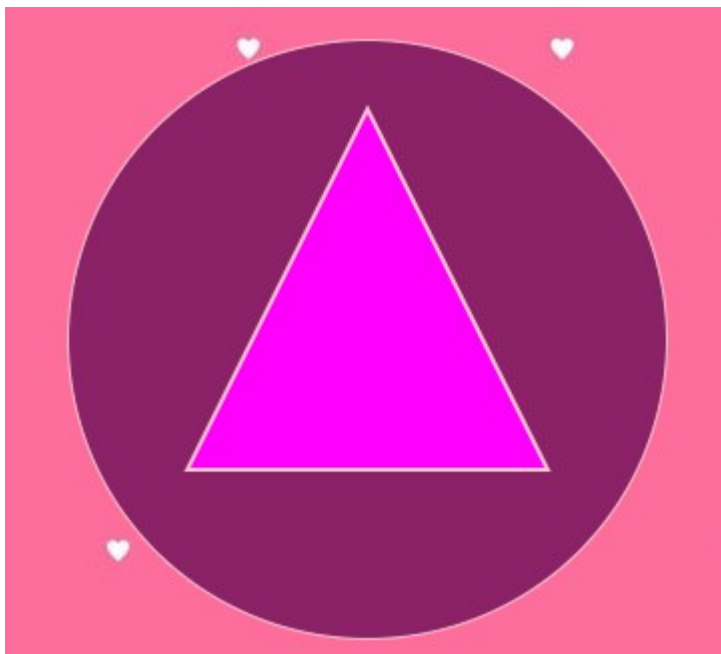
- a. Un triángulo verde.
- b. Un cuadrado rosa.
- c. Un polígono con forma de estrella.
- d. Un cuadrado cuyo relleno cambie de color cada 2 segundos.
- e. Un cuadrado que se mueva en el eje horizontal.
- f. Un texto girado 45°

En este manual lo haremos paso a paso cada uno de ellos, aunque yo lo hago de alguna manera en específica, eso no significa que sea la única manera de hacerlo.

Es decir, cuando yo lo haga con svg es hecho por mi propio criterio

1.4.1 Triangulo

(En mi caso, en el ejercicio tendría que tener fondo verde, pero siguiendo el diseño de la pagina, lo haré rosa)

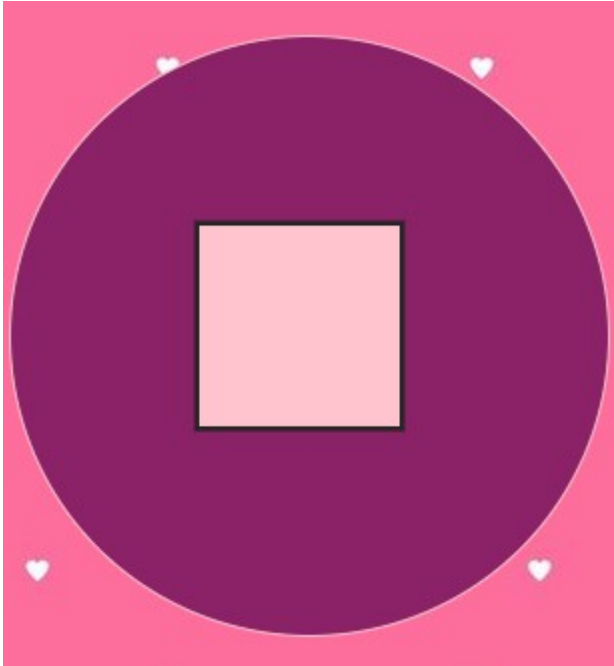


```
<svg height="200" width="200" class="mt-4">  
  <polygon points="100,10 190,190 10,190"  
    stroke="pink" stroke-width=2  
    fill="fuchsia"/>  
</svg>  
<svg width=200 height=200>  
</svg>
```

Gracias a los atributos de stroke, borde, y fill, relleno, definiremos los colores, el tamaño del borde y su color

El atributo polygon definirá que lo que estamos creando es un polígono y por ultimo la etiqueta points indica los puntos de las esquinas del polígono. Por parejas de dos números separados por coma.

Esta vez lo haré con canvas, al contrario que en el caso anterior, a parte de tener en el archivo .html la etiqueta canvas tendremos que crear un archivo js, para definir la figura.



Html:

```
<canvas id="canvas" class="canvas"></canvas>
```

JS:

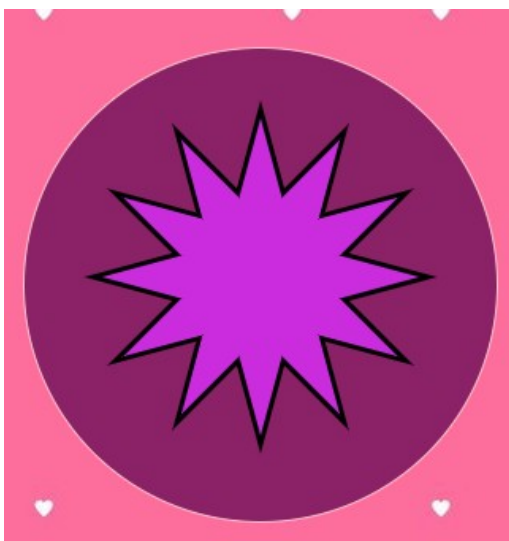
```
const ALTURA_CANVAS = 200,
      ANCHURA_CANVAS = 400;

// Obtener el elemento del DOM
const canvas = document.querySelector("#canvas");
canvas.width = ANCHURA_CANVAS;
canvas.height = ALTURA_CANVAS;
// Del canvas, obtener el contexto para poder dibujar
const contexto = canvas.getContext("2d");
// Estilo de dibujo
// Grosor de línea
contexto.lineWidth = 5;
// Color de línea
contexto.strokeStyle = "#212121";
// Color de relleno
contexto.fillStyle = "pink";
// Dibujamos un rectángulo con lados iguales
contexto.rect(5, 5, 100, 100);
// Hacemos que se dibuje
contexto.stroke();
// Lo rellenamos
contexto.fill();
```

Como ya tengo comentado cada línea de código no me parare a explicarlo en profundidad

1.4.3 Polígono en forma de estrella

En mi caso lo haré con canvas, la razón es que modificando los valores del js podremos modificar la figura como nosotros deseamos.



```
var canvas = document.getElementById("lienzo");
```

```
if (canvas && canvas.getContext) {
```

```
var ctx = canvas.getContext("2d");
```

```
if (ctx) {
```

```
//Fondo y tamaño del borde
```

```
ctx.fillStyle = "rgb(200, 39, 219)";
```

```
ctx.lineWidth = 6;
```

```
//Tendremos que definir la posición en los dos ejes y el del radio
```

```
var X = canvas.width / 2;
```

```
var Y = canvas.height / 2;
```

```
var R = 100;
```

15

```
// Lo mas importante definiremos los lados y el paso
```

```
var L = 12;
```



```
var paso = 5
```

```
//Todos los ángulos centrales de un polígono regular son iguales entre si y su valor en radianes  
puede obtenerse a partir del numero de lados y del paso del polígono
```

```
var estrella= L / paso  
var rad = (2*Math.PI) / estrella;
```

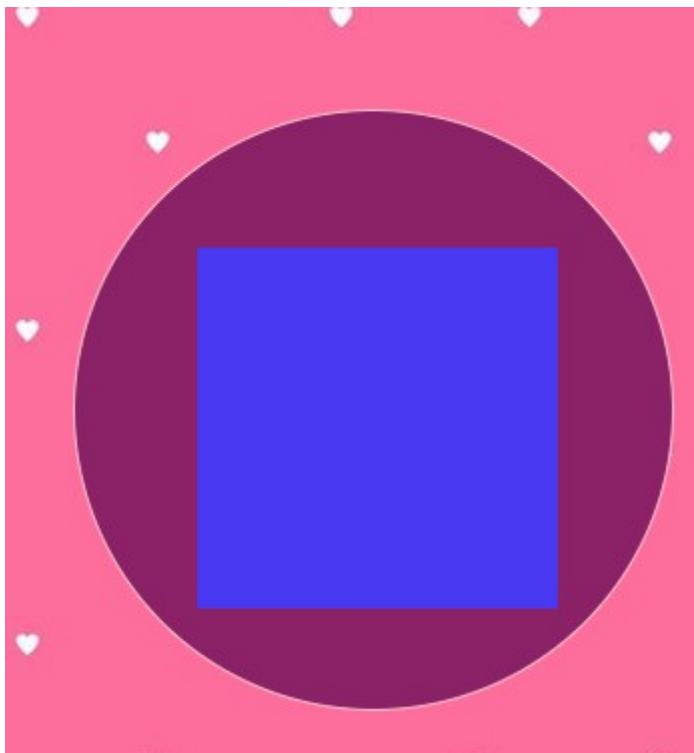
```
ctx.beginPath();
```

```
//Para dibujar el trazado utilizaremos un bucle for donde cada paso del bucle dibujaremos un lado  
de la estrella
```

```
for( var i = 0; i < L; i++ ){  
  x = X + R * Math.cos( rad*i );  
  y = Y + R * Math.sin( rad*i );  
  ctx.lineTo(x, y);  
}  
ctx.closePath();  
ctx.stroke();  
ctx.fill();  
}  
}
```

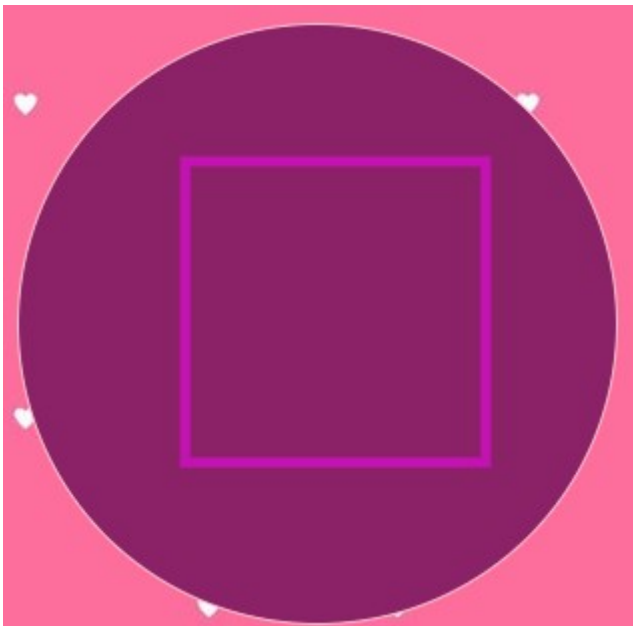
1.4.4 Un cuadrado que su relleno cambie cada 2 segundos

Lo crearemos a partir de un svg, para crear el cuadrado y usar la etiqueta animate para poder darle la animación que queremos realizar



```
<svg width="200" height="200" class="mt-5 mr-3"  
>  
  <rect x="20" y="20" width="250"  
height="250" style="fill:blue">  
  
    //Esta etiqueta es la mas importante, donde  
indicaremos que el fill, es decir el fondo cambiara  
de azul a rosa en un tiempo de 2 segundos  
    <animate attributeType="CSS"  
attributeName="fill" from="blue" to="pink" dur="2s"  
repeatCount="indefinite">  
      </rect>  
    </svg>
```


Al igual que en el caso anterior utilizaremos svg, pero a diferencia de los anteriores tendremos que usar el css, donde utilizaremos @Keyframes



HTML:

```
<svg class="mt-5" width="400" height="180"
id="anim">
  <rect x="50" y="20" width="150"
height="150" style="fill:rgb(133, 29,
97);stroke:fuchsia;stroke-width:5;opacity:0.5" /
  >
  Sorry, your browser does not support
  inline SVG.
</svg>
```

Como podemos ver en el css, deje los comentarios, así que no me repetiré para explicarlos

CSS:

```
/*Movimiento Horizontal del cuadrado hecho con svg*/
@keyframes mianim {
  from { left: 20%;} /*Fotograma inicio*/
  to { left: 10%;} /*Fotograma final*/
}
/*Regla keyframes para Chrome y Safari*/
@-webkit-keyframes mianim {
  from { left: 1%;} /*Fotograma inicio*/
  to { left: 1%;} /*Fotograma final*/
}
/*posicionamiento, anchura, altura y borde elemento.*/
#anim { position: relative ;
}

/*propiedades de animación*/

#anim { /*Forma estándar*/

  animation-name: mianim; /*referencia nombre*/
  animation-duration: 2s; /*tiempo*/
  animation-iteration-count: infinite; /*repetición indefinida*/
  animation-direction: alternate; /*repetir al revés*/

}
```

1.4.5 Texto a 45 grados



En este caso, lo único que tendremos que tener en nuestro HTML sera lo siguiente:

```
<p class="text-center transformed display-1">45°</p>
```

Aunque tendremos que añadir algo en el css, para cambiar la posición de nuestro texto

```
.transformed{
    /* idéntico a rotateZ(45deg); */
    transform: rotate(45deg);
    width: 50px;
    height: 50px;
    margin-left: 42%;
    margin-top: 25%;
}
```

Por ultimo en mi caso en el botón de vuelta a home es un svg en forma de corazón, con una animación que recuerda a las palpitaciones de un corazón.



```
<svg class="heart mr-5 mt-3" viewBox="0 0 32 29.6">
  <path d="M23.6,0c-3.4,0-6.3,2.7-
7.6,5.6C14.7,2.7,11.8,0,8.4,0C3.8,0,0,3.8,0,8.4c0,9.4,9.5,11.9,16,21.2
c6.1-9.3,16-12.1,16-21.2C32,3.8,28.2,0,23.6,0z"/>
</svg>
```

En el css, para que realice la animación habrá que poner lo siguiente:

```
.heart {
  fill: black;
  top: 5px;
  width: 50px;
  animation: pulse 1s ease infinite;
  margin-right: 10%;
}
.heart:hover {
  fill: fuchsia;
}

@keyframes pulse {
  0% { transform: scale(1); }
  50% { transform: scale(1.3); }
  100% { transform: scale(1); }
}
```