

# README

Enlace de la página portada:

<https://viajaporcolombiauoc.netlify.app/portada.html>

Repositorio:

[https://github.com/joalopezmo/PEC1\\_HTML\\_CSSI.git](https://github.com/joalopezmo/PEC1_HTML_CSSI.git)

Para el desarrollo de la segunda actividad se inicio con las pautas de la corrección de la primera PEC.

Primero actualizando parcel a su segunda versión, creando un script de depuración de la carpeta dist a través de rimraf, igualmente instalamos la dependencia jest para pruebas que sin embargo no son del alcance de esta materia; instalamos la dependencia browserslist y forzamos la actualización de la dependencia postcss de parcel a la versión 8.2.7 ya que al actualizar el parcel el script build mandaba error por versión postcss incompatible.

```
EC_I > {} package.json > ...
1  {
2    "name": "pec_i",
3    "version": "1.0.0",
4    "description": "",
5    "scripts": {
6      "build": "parcel build ./src/portada.html",
7      "clean": "rimraf dist",
8      "dev": "parcel src/index.html --open",
9      "format": "prettier --write \"src/**/*.js\"",
10     "lint": "eslint \"src/**/*.js\" --quiet",
11     "test": "jest --passWithNoTests --silent"
12   },
13   "keywords": [],
14   "author": "",
15   "license": "ISC",
16   "devDependencies": {
17     "autoprefixer": "^9.8.6",
18     "browserslist": "^4.18.1",
19     "jest": "^27.4.3",
20     "parcel": "^2.0.1",
21     "parcel-plugin-imagemin": "^4.0.2",
22     "prettier": "^2.4.1",
23     "rimraf": "^3.0.2"
24   },
25   "dependencies": {
26     "postcss": "^8.2.7"
27   }
28 }
```

Se realizaron distintos ejercicios con el objetivo de cumplir con lo requerido en la PEC.



En la pagina medellin.html se creo una imagen clase svg-clipped con la intención de crear una animación basada en 3 círculos con el atributo animate, ubicadas en posición x 50 y 50; x 500 y 20; y en el extremo inferior derecho de la imagen, de radios distintos; todos con una animación repetitiva de 5 seg, con avance de 500 px del radio circular. A continuación el código empleado:

```

<svg height="0" width="0">
  <defs>
    <clipPath id="svgPath">
      <circle
        stroke="#000000"
        stroke-miterlimit="100"
        cx="50"
        cy="50"
        r="40"
      >
      <animate
        attributeName="r"
        attributeType="XML"
        from="0"
        to="500"
        begin="0s"
        dur="5s"
        fill="freeze"
        repeatCount="indefinite"
      />
    </clipPath>
  </defs>
</svg>
```

Para la directa aplicación del estilo declarado en el html en este componente agregamos las clases en archivo css y lo afectamos con el atributo clip-path.

```
.svg-clipped{
  clip-path: url(#svgPath);
  text-align: center;
  width: 100%;
  height: 100%;
}
```

```
.svgenlace{
  text-align: center;
  padding-bottom: 5%;
}
```

Posteriormente aplicamos dirección de arte a las imágenes contenidas en el html según lo requerido en la actividad, mediante la media query max-width 800

```
<picture>
  <source
    media="(max-width:800px)"
    srcset="img/El_Poblado_Medellín2.png"
  />
  
</picture>
```

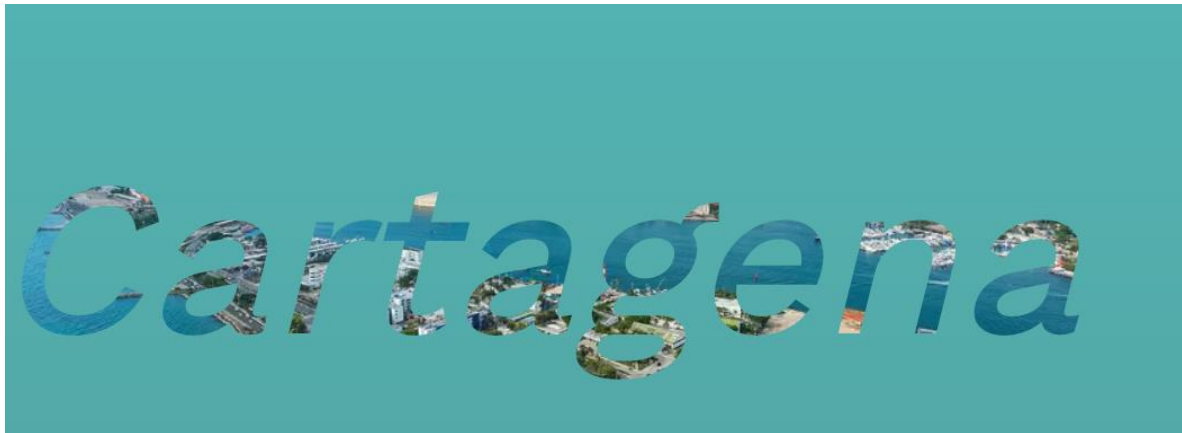
Para el archivo Cartagena.html, nos basamos en la utilización de otro clip-path y de la referencia de un archivo tipo webp un poco mas optimizado según la escala a visualizar, esto para adaptarlo a un vista responsiva.

```

<picture>
  <source
    type="image/webp"
    srcset="
      img/Hyatt-Regency-Cartagena-P016-Bay-View-with-Old-City.800x450.webp
    "
  />
  
  <svg height="0" width="0">
    <defs>
      <clipPath id="svgPath">
        <text
          x="0"
          y="300"
          lengthAdjust="spacing"
          font-size="9.5em"
          font-weight="700"
          font-style="italic"
        >
          Cartagena
        </text>
      </clipPath>
    </defs>
  </svg>
</picture>

```

En la etiqueta picture fue incluida el svg, de forma que sirviera como mascara las letras “Cartagena” a la imagen responsiva que fue tirada atrás.



También dentro de esta pagina me tome la libertad de incluir un par de transiciones que me costaron acoplar la página en específico,



3 transiciones de las fotos contenidas en la parte inferior, solo para un viewport superior a los 768px (ya que no pude hacer que para una resolución menor la transición fuera un poco mas estilizada según la modificación de los atributos de ancho y alto), contenidas en el div clase zoom

```
<h3>Actividades que no te puedes perder</h3>
<div id="container3">
  <div class="zoom">
    
    <p class="text2">
      Conocer la arquitectura del convento del cerro de la Popa
    </p>
  </div>
</div>
```

El estilo aplicado dentro de la media query fue muy sencillo; simplemente tomamos la característica transition, y el estado hover para hacer que la imagen aumentara su tamaño, volviéndola mas opaca, ya que la clase imagen en estado hover la hacia mas trasparente; con el objetivo que la imagen fuera mas visible y mas llamativa.

```
.zoom .imagen{
  /* Aumentamos la anchura y altura durante 2 segundos */
  transition: width 2s, height 2s, transform 2s;
}
.zoom .imagen:hover{
  /* transformamos el elemento al pasar el mouse por encima al doble de su tamaño con scale(2). */
  transform : scale(2);
  opacity: 1;
}
```

Por último se creo un logo, un polígono de ancho y largo 100, para acompañar tanto el header como el footer, y servir de inspiración para el favicon



```
<<span><svg
  version="1.1"
  width="100"
  height="100"
  viewBox="0 0 320 320"
  fill="none"
  stroke="#000"
  stroke-linecap="round"
  xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink"
>
  <defs>
    <path id="r1">
      <animate
        id="p1"
        attributeName="d"
        values="m160,160l0,0 0,0;m130,110l30,-17 30,17;m130,60l30,-17 30,17;m160,20l0,0 0,0"
        dur="6s"
        repeatCount="indefinite"
      />
      <animate
        attributeName="stroke-width"
        values="0;4;4;4;0"
        dur="6s"
        repeatCount="indefinite"
        begin="p1.begin"
      />
    </path>
  </defs>
</svg>
```

Sin duda alguno el atributo values que me permitía definir la animación tanto en la línea 1 como en el stroke-width y como iba a ser trazadas su trayectoria mediante la etiqueta use que me deja rotar la línea mientras está siendo animada fue la peor parte de la realización de esta animación.

```
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
>
  <defs>
    <path id="linea1">
      <animate
        id="p1"
        attributeName="d"
        values="m160,160l0,0 0,0;m130,110l30,-17 30,17;m130,60l30,-17 30,17;m160,20l0,0 0,0"
        dur="6s"
        repeatCount="indefinite"
      />
      <animate
        attributeName="stroke-width"
        values="0;4;4;4;0"
        dur="6s"
        repeatCount="indefinite"
        begin="p1.begin"
      />
    </path>
  </defs>
  <use xlink:href="#linea1" />
  <use xlink:href="#linea1" transform="rotate(60 160 160)" />
  <use xlink:href="#linea1" transform="rotate(120 160 160)" />
  <use xlink:href="#linea1" transform="rotate(180 160 160)" />
  <use xlink:href="#linea1" transform="rotate(240 160 160)" />
  <use xlink:href="#linea1" transform="rotate(300 160 160)" />
</svg>
```

Por último, me gustaría agregar que aunque la animación de la portada inicial fue realizada para pensar en la aplicabilidad javascript, me encuentro trabajando para modificarla y tratar de mejorarla mediante código css

**Jorge Alejandro Lopez Montoya**

Estudiante de la asignatura herramientas HTML y CSS I

Master universitario en desarrollo de sitios y aplicaciones web

E-mail: [joalopezmo@uoc.edu](mailto:joalopezmo@uoc.edu)

Universitat Oberta de Catalunya