



# Universidad Juárez del Estado de Durango

## Facultad de Ingeniería Ciencias y Arquitectura

Ingeniería en Tecnologías Computacionales

Programación Web

Reporte de Práctica 19

Diego Rea Morales

Ing. Fabian Gallegos Gutiérrez

10 de Octubre del 2025

# Índice

Contenido\_\_\_\_\_3

- Explicación del código JavaScript\_\_\_\_\_5

Conclusiones\_\_\_\_\_6

# Contenido

En esta práctica, la idea era crear una pequeña aplicación web interactiva que nos permitiera dibujar polígonos regulares de forma dinámica. El proyecto combina tres tecnologías clave: **HTML** para montar la estructura básica, que incluye un campo para que el usuario ingrese el número de lados, un par de botones y, lo más importante, un elemento `<canvas>`; **CSS** para darle un estilo visual simple y agradable a la interfaz; y **JavaScript**, que es donde ocurre toda la acción, encargándose de tomar el número de lados y realizar el dibujo sobre el lienzo (canvas). La funcionalidad es directa: pones un número, le das al botón "Dibujar" y la figura aparece.

```
index.html styles.css script.js M
Practica19 > index.html > html > body > div.dibujo
1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <meta charset="UTF-8" />
6
7      <title>Polígono</title>
8
9      <link rel="stylesheet" type="text/css" href="styles.css" />
10   </head>
11
12   <body>
13     <form>
14       <label for="lados">Número de lados:</label>
15
16       <input type="number" id="lados" name="lados" min="3" max="20" />
17
18       <button type="button" onclick="dibujarPoligono()">Dibujar</button>
19
20       <button type="button" onclick="dibujarRecta()">Recta</button>
21       <label for="angulos" id="angle"></label>
22     </form>
23
24     <div class="dibujo">
25       <canvas id="canvas"></canvas>
26     </div>
27
28     <script src="script.js"></script>
29   </body>
30 </html>
31
```

```
index.html styles.css scriptjs M
Practica19 > styles.css > button
1 body {
2   background-color: #333;
3   align-items: center;
4   justify-content: center;
5   font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
6   font-weight: bold;
7 }
8 form {
9   background-color: rgba(255, 255, 255, 0.5);
10  border-radius: 10px;
11  box-shadow: 0px 0px 10px 0px rgba(255, 255, 255, 0.5);
12  padding: 20px;
13  margin: 20px;
14  text-align: center;
15  align-items: center;
16  justify-content: center;
17 }
18 button {
19   background-color: #fff;
20   border-radius: 5px;
21   border: 1px solid #ddd;
22   color: #333;
23   cursor: pointer;
24   font-size: 16px;
25   margin-top: 10px;
26   padding: 10px;
27   transition: all 0.3s ease;
28 }
29 button:hover {
30   background-color: #333;
31   border-color: #333;
32   color: #fff;
33 }
34 .dibujo {
35   display: flex;
36   align-items: center;
37   justify-content: center;
38   flex: auto;
39 }
40 #canvas {
41   width: 100%;
42   height: 50%;
43   background-color: #fff;
44 }
```

```
index.html styles.css scriptjs M
Practica19 > scriptjs > dibujarRecta
1 function dibujarPoligono() {
2   const canvas = document.getElementById("canvas");
3   const lados = document.getElementById("lados").value;
4   const ctx = canvas.getContext("2d");
5   const centroX = canvas.width / 2;
6   const centroY = canvas.height / 2;
7   const radio = 70;
8   const angulo = (2 * Math.PI) / lados;
9
10  ctx.clearRect(0, 0, canvas.width, canvas.height);
11  ctx.beginPath();
12  ctx.moveTo(centroX + radio * Math.cos(0), centroY + radio * Math.sin(0));
13
14  for (let i = 1; i <= lados; i++) {
15    const x = centroX + radio * Math.cos(i * angulo);
16    const y = centroY + radio * Math.sin(i * angulo);
17    ctx.lineTo(x, y);
18  }
19  let angulos = document.getElementById("angle");
20  angulos.innerHTML =
21    lados == 0 ? "" : "Valor de los Angulos: " + (180 * (lados - 2)) / lados;
22
23  ctx.closePath();
24  ctx.stroke();
25 }
26 function dibujarRecta() {
27   const canvas = document.getElementById("canvas");
28   const ctx = canvas.getContext("2d");
29
30   const recta = canvas.width * 0.4;
31   const centroX = canvas.width / 2;
32   const centroY = canvas.height / 2;
33
34   const inicio = centroX - (recta / 2);
35   const fin = centroX + (recta / 2);
36
37   ctx.beginPath();
38   ctx.moveTo(inicio, centroY);
39   ctx.lineTo(fin, centroY);
40   ctx.closePath();
41   ctx.stroke();
42 }
```

## Explicación del código JavaScript

La magia de verdad ocurre en el archivo `script.js`, principalmente dentro de la función `dibujarPoligono()`. Cuando la llamas al presionar el botón, lo primero que hace es tomar el número de lados que el usuario escribió. Luego, calcula el ángulo que debe haber entre cada vértice del polígono dividiendo los 360 grados de un círculo (expresado en radianes,  $2 * \text{Math.PI}$ ) entre el número de lados. Usando un bucle `for` y un poco de trigonometría (con `Math.cos` y `Math.sin`), va calculando las coordenadas X y Y de cada uno de los vértices alrededor de un punto central en el `canvas`. Conforme calcula cada punto, traza una línea hasta él con `ctx.lineTo()`. Al final del bucle, simplemente cierra la figura y la dibuja con `ctx.stroke()`.

Adicionalmente, calcula el valor de los ángulos internos del polígono y lo muestra en pantalla. La otra función, `dibujarRecta()`, es mucho más sencilla y sirve como un ejemplo básico de cómo trazar una simple línea en el `canvas`.

## Conclusiones

Para terminar, esta práctica fue una probadita genial de lo poderoso que es el elemento `<canvas>` de HTML cuando se combina con JavaScript. Sirvió para entender que podemos crear gráficos y figuras dinámicas directamente en el navegador sin depender de imágenes externas. Lo más interesante fue ver cómo las matemáticas, en este caso la geometría y trigonometría, se dan la mano con la programación para lograr un resultado visual. Entender cómo un círculo y sus ángulos pueden usarse para posicionar los vértices de un polígono es la base para cosas mucho más complejas. Sin duda, es un excelente punto de partida para quien quiera meterse a temas más avanzados como crear visualizaciones de datos, animaciones o incluso videojuegos sencillos para la web.