

# Juego 2D desarrollado en JavaScript puro: Introducción a la Informática

**JHON ESTEBAN RESTREPO ESCOBAR**  
SEPTIEMBRE DE 2020



# 1 CONTENIDO

---

1	CONTENIDO.....	1
2	PRESENTACIÓN.....	2
3	FASE 1: Dibujar y mover una bola .....	3
4	FASE 2: Rebotando en las paredes.....	6
5	FASE 3: Control de la pala y el teclado .....	9
6	FASE 4: Fin del juego .....	13
7	FASE 5: Muro de ladrillos .....	17
8	FASE 6: Detección de colisiones .....	22
9	FASE 7: Contar puntos y ganar .....	27
10	FASE 8: Controlando el ratón .....	32
11	FASE 9: Finalizando el juego.....	38
12	CONCLUSIONES .....	44
13	BIBLIOGRAFÍA.....	45

## 2 PRESENTACIÓN

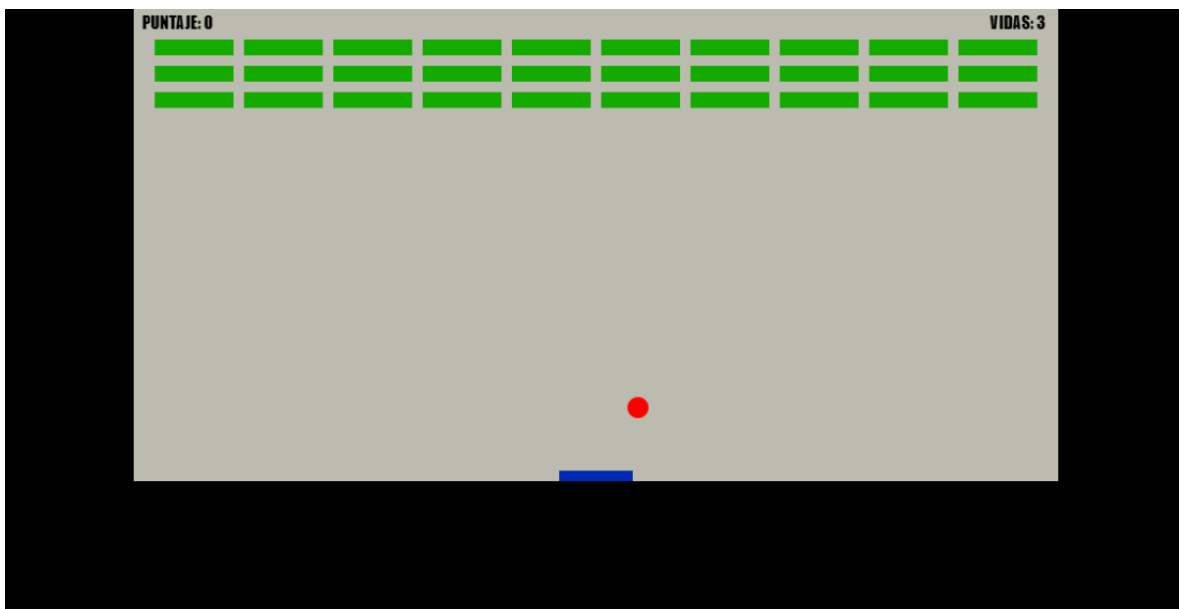
---

La presente monografía describe el desarrollo metódico de un juego 2D elaborado utilizando HTML5, CSS, CANVAS y JavaScript.

El juego elaborado se crea con JavaScript puro, utilizando un enfoque metódico en el cual se avanza de versión en versión, de modo que cada nuevo programa abarca un aspecto adicional del juego.

En cada fase del juego es una consecución de la anterior donde paso a paso se le va dando forma al juego en 2D. Donde se va a poder visualizar el código del programa y después de este la ejecución del mismo.

Al finalizar el juego se va tener un juego clásico en 2D con demasiados conceptos que pueden ser usados en un futuro para diferentes proyectos realizados en la web.



Gráfica 1. Juego 2D en JavaScript

El documento web que sirve como referencia para el desarrollo del juego está en el siguiente enlace:

[https://developer.mozilla.org/es/docs/Games/Workflows/Famoso\\_juego\\_2D\\_usando\\_JavaScript\\_puro](https://developer.mozilla.org/es/docs/Games/Workflows/Famoso_juego_2D_usando_JavaScript_puro)

**AUTOR: JHON ESTEBAN RESTREPO ESCOBAR**



### 3 FASE 1: DIBUJAR Y MOVER UNA BOLA

---

El primer paso consiste en elaborar una página HTML básica. Agregaremos a dicha página un elemento CANVAS, el cual nos servirá como base para el desarrollo del juego 2D.

El código JavaScript que operará sobre el CANVAS debe encerrarse entre las etiquetas `<script>...</script>`

La correcta visualización del CANVAS requiere de la adición de algunas características de estilo. Una vez hecho esto, se procede a establecer la codificación pertinente del JavaScript. Debe notarse la inclusión de algunas variables que definen la funcionalidad del juego en sus aspectos básicos: las coordenadas en las que se encuentra la bola y los valores de incremento para modificar su posición.

Se definen tres funciones importantes. La primera de ellas, `dibujarBola ()`, se encarga de dibujar sobre la pantalla una bola con el color indicado en los estilos. La segunda función se denomina `dibujar ()`, y es la encargada de limpiar el CANVAS, dibujar la bola y cambiar los valores de las coordenadas. Finalmente, la función `setInterval(dibujar, 40)`, llama a la función `dibujar` cada 40 milisegundos.

También definimos el fondo negro, el cual codificamos a través del canvas donde el programa lo que hace es a partir del borde de este, toma todo lo de afuera y lo pone en el color que deseado en este caso negro.

El código fuente del programa es el siguiente (para darle formato, se deben seguir las instrucciones disponibles en: <https://trabajonomada.com/insertar-codigo-word/> y seguidamente utilizar el enlace: [http://qbnz.com/highlighter/php\\_highlighter.php](http://qbnz.com/highlighter/php_highlighter.php))

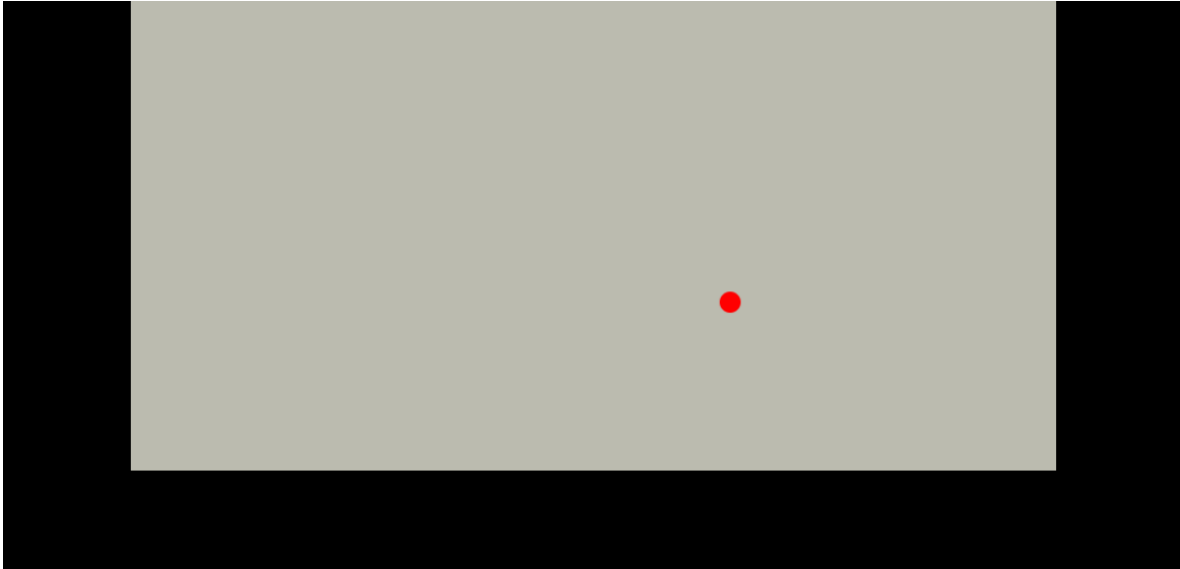
```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.
6.     <title>JuegoP1</title>
7.
8.     <style>
9.         * { padding: 0; margin: 0;
10.        }
11.        canvas { background: #BBBBAF; display: block;
margin: 0 auto;
12.        }
13.
14.     </style>
15.
```

```

16.     </head>
17.
18.     <body style="background-color: #000000;">
19.
20.     <canvas
id="miCanvas" width="880" height="450" ></canvas>
21.
22.     <script>
23.         var canvas = document.getElementById("miCanvas");
24.         var ctx = canvas.getContext("2d");
25.         var x = canvas.width/2;
26.         var y = canvas.height-30;
27.         var dx = 10;
28.         var dy = -10;
29.
30.         function dibujarBola() {
31.
32.             ctx.beginPath();
33.             ctx.arc(x, y, 10, 0, Math.PI*2);
34.             ctx.fillStyle = "#FF0000";
35.             ctx.fill();
36.             ctx.closePath();
37.         }
38.
39.         function dibujar() {
40.
41.             ctx.clearRect(0, 0, canvas.width, canvas.height)
42.             ;
43.             dibujarBola();
44.             x = x + dx;
45.             y = y + dy;
46.         }
47.         setInterval(dibujar, 40);
48.     </script>
49. </body>
50. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



### Gráfica 2. La interfaz inicial del juego

En la gráfica 2 se aprecia el dibujo de la bola, y la secuencia de movimiento a partir de los incrementos en X y Y que fueron definidos.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 4 FASE 2: REBOTANDO EN LAS PAREDES

---

El segundo paso es crear los límites dentro de nuestro lienzo para que la bola rebote dentro de este.

En este paso se crean dos condiciones las cuales generan los límites permitidos a los que la bola puede llegar y va a rebotar:

La primera condición es `if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) { dx = -dx;}`, esta condición crea el rango horizontal al que la pelota se puede desplazar.

La segunda condición es `if(y + dy > canvas.height-ballRadius || y + dy < ballRadius) {dy = -dy;}`, esta condición crea el rango vertical al que la pelota se puede desplazar.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>JuegoP2</title>
6.
7.     <style>
8.         * { padding: 0; margin: 0; }
9.
10.         canvas { background: #BBBBAF; display: block; margin: 0 auto; }
11.
12.     </style>
13.
14. </head>
15.
16. <body style="background-color: #000000;">
17.
18.     <canvas
19.     id="miCanvas" width="880" height="450"> </canvas>
20.
21.     <script>
22.
23.         var canvas = document.getElementById("miCanvas");
24.
25.         var ctx = canvas.getContext("2d");
26.         var ballRadius = 10;
27.         var x = canvas.width / 2;
28.         var y = canvas.height - 30;
29.         var dx = 10;
30.         var dy = -10;
```

```

30.         function dibujarBola() {
31.
32.             ctx.beginPath();
33.             ctx.arc(x, y, ballRadius, 0, Math.PI*2);
34.             ctx.fillStyle = "#ff0000";
35.             ctx.fill();
36.             ctx.closePath();
37.         }
38.
39.         function dibujar() {
40.
41.             ctx.clearRect(0, 0, canvas.width, canvas
42.             .height);
43.
44.             dibujarBola();
45.
46.             if (x + dx > canvas.width -
47.             ballRadius || x + dx < ballRadius) {
48.                 dx = dx * (-1);
49.             }
50.
51.             if(y + dy > canvas.height -
52.             ballRadius || y + dy < ballRadius) {
53.                 dy = dy * (-1);
54.             }
55.
56.             x = x + dx;
57.             y = y + dy;
58.
59.             setInterval(dibujar, 40);
60.         }
61.     }
62. }
63.
64. </script>
65. </body>
66. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:





Gráfica 3. Ejecución del segundo programa.

En la figura 3 podemos observar a la bola rebotando y cumpliendo con los límites anteriormente definidos en las condiciones.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 5 FASE 3: CONTROL DE LA PALETA Y EL TECLADO

---

El paso numero 3 consiste en crear la paleta en la cual la bola rebotará, la cual estará situada en la parte inferior del juego y será controlada por ambas flechas del teclado.

Para empezar con el tercer paso primero se deben crear dos variables a las cuales se les asignara el movimiento de la paleta con las flechas del teclado. Estas variables llevaran el siguiente nombre flechaDerechaPulsada y flechalzquierdaPulsada. Luego de esto se crea una función la cual maneja el movimiento de la tecla presionada y otro de la tecla liberada.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>JuegoP3</title>
6.     <style>
7.         * {padding: 0;margin: 0;
8.
9.     }
10.        canvas {background: #BBBBAF;display: block;margi
n: 0 auto;
11.        }
12.    </style>
13.
14. </head>
15.
16. <body style="background-color: #000000;">
17.
18. <canvas
id="miCanvas" width="880" height="450" ></canvas>
19.
20. <script>
21.     var canvas = document.getElementById("miCanvas");
22.     var ctx = canvas.getContext("2d");
23.
24.     var radioBola = 10;
25.     var x = canvas.width/2;
26.     var y = canvas.height-30;
27.     var dx = 10;
28.     var dy = -10;
29.
30.     var alturaPaleta = 10;
31.     var anchuraPaleta = 70;
32.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
33.
34.     var flechaDerechaPulsada = false;
```

```

35.         var flechaIzquierdaPulsada = false;
36.
37.         document.addEventListener("keydown", manejadorTeclaP
resionada, false);
38.         document.addEventListener("keyup", manejadorTeclaLib
erada, false);
39.
40.         function manejadorTeclaPresionada(e) {
41.             if(e.keyCode == 39) {
42.                 flechaDerechaPulsada = true;
43.             }
44.             else if(e.keyCode == 37) {
45.                 flechaIzquierdaPulsada = true;
46.             }
47.         }
48.
49.         function manejadorTeclaLiberada(e) {
50.             if(e.keyCode == 39) {
51.                 flechaDerechaPulsada = false;
52.             }
53.             else if(e.keyCode == 37) {
54.                 flechaIzquierdaPulsada = false;
55.             }
56.         }
57.
58.         function dibujarBola() {
59.             ctx.beginPath();
60.             ctx.arc(x, y, radioBola, 0, Math.PI*2);
61.             ctx.fillStyle = "#ff0000";
62.             ctx.fill();
63.             ctx.closePath();
64.         }
65.
66.         function dibujarPaleta() {
67.             ctx.beginPath();
68.             ctx.rect(paletaPosX, canvas.height-
alturaPaleta, anchuraPaleta, alturaPaleta);
69.             ctx.fillStyle = "#0029B3";
70.             ctx.fill();
71.             ctx.closePath();
72.         }
73.         function dibujar() {
74.             ctx.clearRect(0, 0, canvas.width, canvas.height)
;
75.             dibujarBola();
76.             dibujarPaleta();
77.
78.             if(x + dx > canvas.width-
radioBola || x + dx < radioBola) {

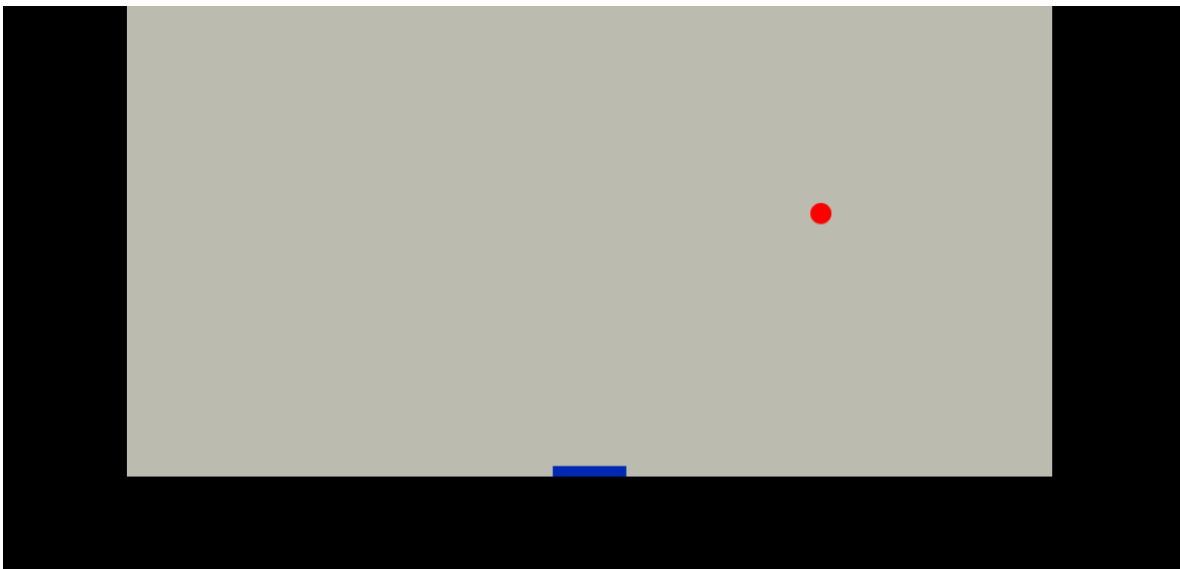
```

```

79.         dx = -dx;
80.     }
81.     if(y + dy > canvas.height-
radioBola || y + dy < radioBola) {
82.         dy = -dy;
83.     }
84.
85.     if(flechaDerechaPulsada && paletaPosX < canvas.w
idth-anchuraPaleta) {
86.         paletaPosX += 7;
87.     }
88.     else if(flechaIzquierdaPulsada && paletaPosX > 0
) {
89.         paletaPosX -= 7;
90.     }
91.
92.     x += dx;
93.     y += dy;
94. }
95.
96.     setInterval(dibujar, 40);
97. </script>
98. </body>
99. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 4. Ejecución del tercer programa.



En la figura 4 podemos observar la bola y la paleta en la parte inferior del juego creadas anteriormente en la parte número 3 del código.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 6 FASE 4: FIN DEL JUEGO

---

En esta parte del programa programaremos que se pueda detectar cuando la bola toca la base de la pantalla, en una coordenada diferente a la de donde se encuentra la paleta, lo que hará que el juego se pierda.

Para este caso analizaremos un código en la función dibujar, el código sería:  $(y + dy > \text{canvas.height} - \text{radioBola})$  el cual se utilizaría para cuando la bola toque la parte inferior del juego lo cual haría que el juego se pierda. Pero para estar seguros de que el juego se ha perdido analizaremos el siguiente código:  $(x > \text{paletaPosX} \ \&\& \ x < \text{paletaPosX} + \text{anchuraPaleta})$  el cual hace que se analice la posición de la bola y en caso de que lo bola toque la parte inferior hace que se detenga el ciclo de animación del juego y se pierda.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.
6.     <title>JuegoP4</title>
7.
8.     <style>* { padding: 0; margin: 0; } canvas { background:
#BBBBAF; display: block; margin: 0 auto; }</style>
9.
10.    </head>
11.
12.    <body style="background-color: #000000;">
13.
14.    <canvas id="miCanvas" width="880" height="450"></canvas>
15.
16.    <script>
17.        var canvas = document.getElementById("miCanvas");
18.        var ctx = canvas.getContext("2d");
19.
20.        var radioBola = 10;
21.        var x = canvas.width/2;
22.        var y = canvas.height-30;
23.        var dx = 10;
24.        var dy = -10;
25.
26.        var alturaPaleta = 10;
27.        var anchuraPaleta = 70;
28.        var paletaPosX = (canvas.width-anchuraPaleta)/2;
29.
30.        var flechaDerechaPresionada = false;
31.        var flechaIzquierdaPresionada = false;
```

```

32.
33.     document.addEventListener("keydown", manejadorTeclaP
resionada, false);
34.     document.addEventListener("keyup", manejadorTeclaLib
erada, false);
35.
36.     function manejadorTeclaPresionada(e) {
37.         if(e.keyCode == 39) {
38.             flechaDerechaPresionada = true;
39.         }
40.         else if(e.keyCode == 37) {
41.             flechaIzquierdaPresionada = true;
42.         }
43.     }
44.     function manejadorTeclaLiberada(e) {
45.         if(e.keyCode == 39) {
46.             flechaDerechaPresionada = false;
47.         }
48.         else if(e.keyCode == 37) {
49.             flechaIzquierdaPresionada = false;
50.         }
51.     }
52.
53.     function dibujarBola() {
54.         ctx.beginPath();
55.         ctx.arc(x, y, radioBola, 0, Math.PI*2);
56.         ctx.fillStyle = "#FF0000";
57.         ctx.fill();
58.         ctx.closePath();
59.     }
60.     function dibujarPaleta() {
61.         ctx.beginPath();
62.         ctx.rect(paletaPosX, canvas.height-
alturaPaleta, anchuraPaleta, alturaPaleta);
63.         ctx.fillStyle = "#0029B3";
64.         ctx.fill();
65.         ctx.closePath();
66.     }
67.
68.     function dibujar() {
69.         ctx.clearRect(0, 0, canvas.width, canvas.height)
;
70.
71.         dibujarBola();
72.         dibujarPaleta();
73.
74.         if(x + dx > canvas.width-
radioBola || x + dx < radioBola) {
75.             dx = -dx;

```

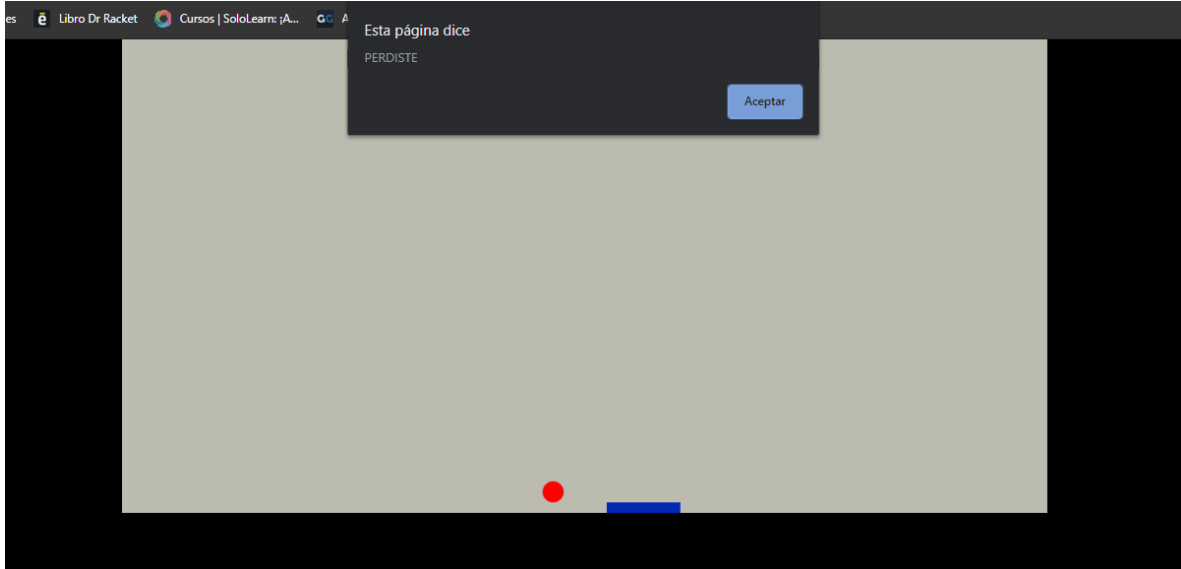
```

76.     }
77.     if(y + dy < radioBola) {
78.         dy = -dy;
79.     }
80.     else if(y + dy > canvas.height-radioBola) {
81.         if(x > paletaPosX && x < paletaPosX + anchur
aPaleta) {
82.             dy = -dy;
83.         }
84.         else {
85.             clearInterval(juego);
86.             alert("PERDISTE");
87.             document.location.reload();
88.         }
89.     }
90.
91.     if(flechaDerechaPresionada && paletaPosX < canva
s.width-anchuraPaleta) {
92.         paletaPosX += 7;
93.     }
94.     else if(flechaIzquierdaPresionada && paletaPosX
> 0) {
95.         paletaPosX -= 7;
96.     }
97.
98.     x += dx;
99.     y += dy;
100.    }
101.
102.    var juego = setInterval(dibujar, 40);
103. </script>
104.
105. </body>
106. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:





Gráfica 5. Ejecución del cuarto programa.

En la figura 5 podemos observar como la bola al tocar la parte inferior del juego y al estar en una coordenada diferente a la paleta aparece un PERDISTE que significa que el juego se ha perdido y se ha acabado

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 7 FASE 5: MURO DE LADRILLOS

---

En esta parte del juego crearemos unas variables las cuales crearan un muro de ladrillos dentro del juego en los cuales rebotara la bola.

Analizaremos la function dibujarLadrillos() , esta función se apoya de varias variables para la creación del muro de los ladrillos la cual la hace analizando la columna y la fila en la que quedara asignado cada ladrillo.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>JuegoP5</title>
6.     <style>* { padding: 0; margin: 0; } canvas { background:
    #BBBBAF; display: block; margin: 0 auto; }</style>
7. </head>
8. <body style="background-color: #000000;">
9.
10.     <canvas id="miCanvas" width="880" height="450"></canvas>
11.
12.     <script>
13.         var canvas = document.getElementById("miCanvas");
14.         var ctx = canvas.getContext("2d");
15.         var radioBola = 10;
16.         var x = canvas.width/2;
17.         var y = canvas.height-30;
18.         var dx = 10;
19.         var dy = -10;
20.
21.         var alturaPaleta = 10;
22.         var anchuraPaleta = 70;
23.         var paletaPosX = (canvas.width-anchuraPaleta)/2;
24.         var flechaDerechaPresionada = false;
25.         var flechaIzquierdaPresionada = false;
26.
27.         var nroFilasLadrillos = 10;
28.         var nroColumnasLadrillos = 3;
29.         var anchoLadrillo = 75;
30.         var alturaLadrillo = 15;
31.         var rellenoLadrillo = 10;
32.         var vacioSuperiorLadrillo = 30;
33.         var vacioIzquierdoLadrillo = 20;
34.         var ladrillos = [];
35.
```

```

36.         for(var columna=0; columna<nroColumnasLadrillos; col
umna++) {
37.             ladrillos[columna] = [];
38.
39.             for(var fila=0; fila<nroFilasLadrillos; fila++)
{
40.                 ladrillos[columna][fila] = { x: 0, y: 0 };
41.             }
42.         }
43.
44.         document.addEventListener("keydown", manejadorTeclaP
resionada, false);
45.         document.addEventListener("keyup", manejadorTeclaLib
erada, false);
46.
47.         function manejadorTeclaPresionada(e) {
48.             if(e.keyCode == 39) {
49.                 flechaDerechaPresionada = true;
50.             }
51.             else if(e.keyCode == 37) {
52.                 flechaIzquierdaPresionada = true;
53.             }
54.         }
55.         function manejadorTeclaLiberada(e) {
56.             if(e.keyCode == 39) {
57.                 flechaDerechaPresionada = false;
58.             }
59.             else if(e.keyCode == 37) {
60.                 flechaIzquierdaPresionada = false;
61.             }
62.         }
63.
64.         function dibujarBola() {
65.             ctx.beginPath();
66.             ctx.arc(x, y, radioBola, 0, Math.PI*2);
67.             ctx.fillStyle = "#FF0000";
68.             ctx.fill();
69.             ctx.closePath();
70.         }
71.
72.         function dibujarPaleta() {
73.             ctx.beginPath();
74.             ctx.rect(paletaPosX, canvas.height-
alturaPaleta, anchuraPaleta, alturaPaleta);
75.             ctx.fillStyle = "#0029B3";
76.             ctx.fill();
77.             ctx.closePath();
78.         }
79.

```

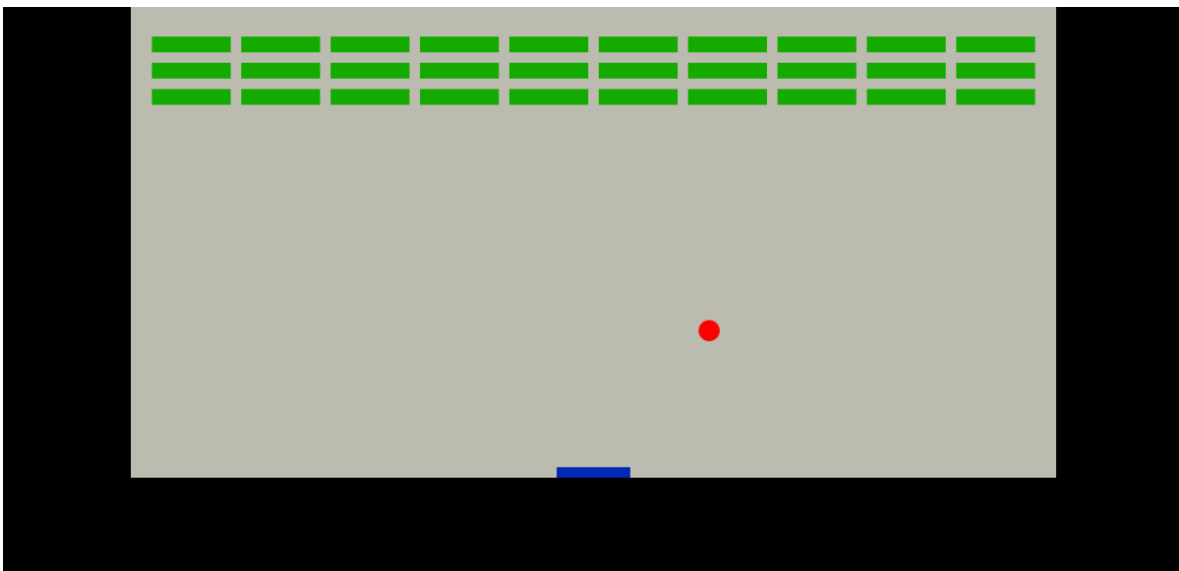
```

80.         function dibujarLadrillos() {
81.             for(var columna=0; columna<nroColumnasLadrillos;
            columna++) {
82.                 for(var fila=0; fila<nroFilasLadrillos; fila
            ++)) {
83.
84.                     var brickX = (fila*(anchoLadrillo+rellen
            oLadrillo))+vacioIzquierdoLadrillo;
85.
86.                     var brickY = (columna*(alturaLadrillo+re
            llenadoLadrillo))+vacioSuperiorLadrillo;
87.
88.                     ladrillos[columna][fila].x = brickX;
89.
90.                     ladrillos[columna][fila].y = brickY;
91.
92.                     ctx.beginPath();
93.                     ctx.rect(brickX, brickY, anchoLadrillo,
            alturaLadrillo);
94.                     ctx.fillStyle = "#15AA00";
95.                     ctx.fill();
96.                     ctx.closePath();
97.                 }
98.             }
99.         }
100.
101.         function dibujar() {
102.             ctx.clearRect(0, 0, canvas.width, canvas.height)
            ;
103.             dibujarLadrillos();
104.             dibujarBola();
105.             dibujarPaleta();
106.
107.             if(x + dx > canvas.width-
            radioBola || x + dx < radioBola) {
108.                 dx = -dx;
109.             }
110.             if(y + dy < radioBola) {
111.                 dy = -dy;
112.             }
113.             else if(y + dy > canvas.height-radioBola) {
114.                 if(x > paletaPosX && x < paletaPosX + anchur
            aPaleta) {
115.                     dy = -dy;
116.                 }
117.                 else {
118.                     clearInterval(juego);
119.                     alert("PERDISTE");
120.                     document.location.reload();

```

```
121.         }
122.     }
123.
124.     if(flechaDerechaPresionada && paletaPosX < canva
s.width-anchuraPaleta) {
125.         paletaPosX += 7;
126.     }
127.     else if(flechaIzquierdaPresionada && paletaPosX
> 0) {
128.         paletaPosX -= 7;
129.     }
130.
131.     x += dx;
132.     y += dy;
133. }
134.
135.     var juego = setInterval(dibujar, 40);
136. </script>
137.
138. </body>
139. </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 6. Ejecución del quinto programa.



En la figura 6 podemos observar la creación de la pared de ladrillos dentro del campo del juego.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 8 FASE 6: DETECCIÓN DE COLISIONES

---

En esta parte del programa realizaremos la función que hará que se detecte la colisión de la bola con alguno de los ladrillos y al ocurrir esto hará que el ladrillo con el que colisiono desaparezca.

Procederemos a analizar la función que hace esto posible, la función será la siguiente: `function deteccionColision()`, esta es la función que permite que cuando la bola colisione con alguno de los ladrillos desaparezca, esto se realiza creando una variable temporal en la cual se asigna el ladrillo y analizando su columna y su fila y así saber si fue impactado.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta charset="utf-8" />
5.   <title>JuegoP6</title>
6.   <style>* { padding: 0; margin: 0; } canvas { background:
   #BBBBAF; display: block; margin: 0 auto; }</style>
7. </head>
8. <body style="background-color: #000000;">
9.
10.   <canvas id="miCanvas" width="880" height="450"></canvas>
11.
12.   <script>
13.     var canvas = document.getElementById("miCanvas");
14.     var ctx = canvas.getContext("2d");
15.
16.     var radioBola = 10;
17.     var x = canvas.width/2;
18.     var y = canvas.height-30;
19.     var dx = 10;
20.     var dy = -10;
21.
22.     var alturaPaleta = 10;
23.     var anchuraPaleta = 70;
24.     var paletaPosX = (canvas.width-anchuraPaleta)/2;
25.
26.     var flechaDerechaPresionada = false;
27.     var flechaIzquierdaPresionada = false;
28.
29.     var nroFilasLadrillos = 10;
30.     var nroColumnasLadrillos = 3;
31.     var anchuraLadrillo = 75;
32.     var alturaLadrillo = 15;
33.     var rellenoLadrillo = 10;
```

```

34.     var vacioSuperiorLadrillo = 30;
35.     var vacioIzquierdoLadrillo = 20;
36.
37.     var ladrillos = [];
38.     for(var c=0; c<nroColumnasLadrillos; c++) {
39.         ladrillos[c] = [];
40.         for(var f=0; f<nroFilasLadrillos; f++) {
41.             ladrillos[c][f] = { x: 0, y: 0, status: 1 };
42.         }
43.     }
44.
45.     document.addEventListener("keydown", manejadorTeclaP
resionada, false);
46.     document.addEventListener("keyup", manejadorTeclaLib
erada, false);
47.
48.     function manejadorTeclaPresionada(e) {
49.         if(e.keyCode == 39) {
50.             flechaDerechaPresionada = true;
51.         }
52.         else if(e.keyCode == 37) {
53.             flechaIzquierdaPresionada = true;
54.         }
55.     }
56.
57.     function manejadorTeclaLiberada(e) {
58.         if(e.keyCode == 39) {
59.             flechaDerechaPresionada = false;
60.         }
61.         else if(e.keyCode == 37) {
62.             flechaIzquierdaPresionada = false;
63.         }
64.     }
65.
66.     function deteccionColision() {
67.
68.         for(var c=0; c<nroColumnasLadrillos; c++) {
69.             for(var f=0; f<nroFilasLadrillos; f++) {
70.                 var b = ladrillos[c][f];
71.                 if(b.status == 1) {
72.                     if(x > b.x && x < b.x+anchuraLadrill
o && y > b.y && y < b.y+alturaLadrillo) {
73.                         dy = -dy;
74.                         b.status = 0;
75.                     }
76.                 }
77.             }
78.         }
79.     }

```



```

80.
81.     function dibujarBola() {
82.         ctx.beginPath();
83.         ctx.arc(x, y, radioBola, 0, Math.PI*2);
84.         ctx.fillStyle = "#FF0000";
85.         ctx.fill();
86.         ctx.closePath();
87.     }
88.
89.     function dibujarPaleta() {
90.         ctx.beginPath();
91.         ctx.rect(paletaPosX, canvas.height-
alturaPaleta, anchuraPaleta, alturaPaleta);
92.         ctx.fillStyle = "#0029B3";
93.         ctx.fill();
94.         ctx.closePath();
95.     }
96.
97.     function dibujarLadrillos() {
98.         for(var c=0; c<nroColumnasLadrillos; c++) {
99.             for(var f=0; f<nroFilasLadrillos; f++) {
100.                 if(ladrillos[c][f].status == 1) {
101.                     var brickX = (f*(anchuraLadrillo+rel
lenoLadrillo))+vacioIzquierdoLadrillo;
102.                     var brickY = (c*(alturaLadrillo+rell
enoLadrillo))+vacioSuperiorLadrillo;
103.                     ladrillos[c][f].x = brickX;
104.                     ladrillos[c][f].y = brickY;
105.                     ctx.beginPath();
106.                     ctx.rect(brickX, brickY, anchuraLadr
illo, alturaLadrillo);
107.                     ctx.fillStyle = "#15AA00";
108.                     ctx.fill();
109.                     ctx.closePath();
110.                 }
111.             }
112.         }
113.     }
114.
115.     function dibujar() {
116.         ctx.clearRect(0, 0, canvas.width, canvas.height)
;
117.         dibujarLadrillos();
118.         dibujarBola();
119.         dibujarPaleta();
120.         deteccionColision();
121.
122.         if(x + dx > canvas.width-
radioBola || x + dx < radioBola) {

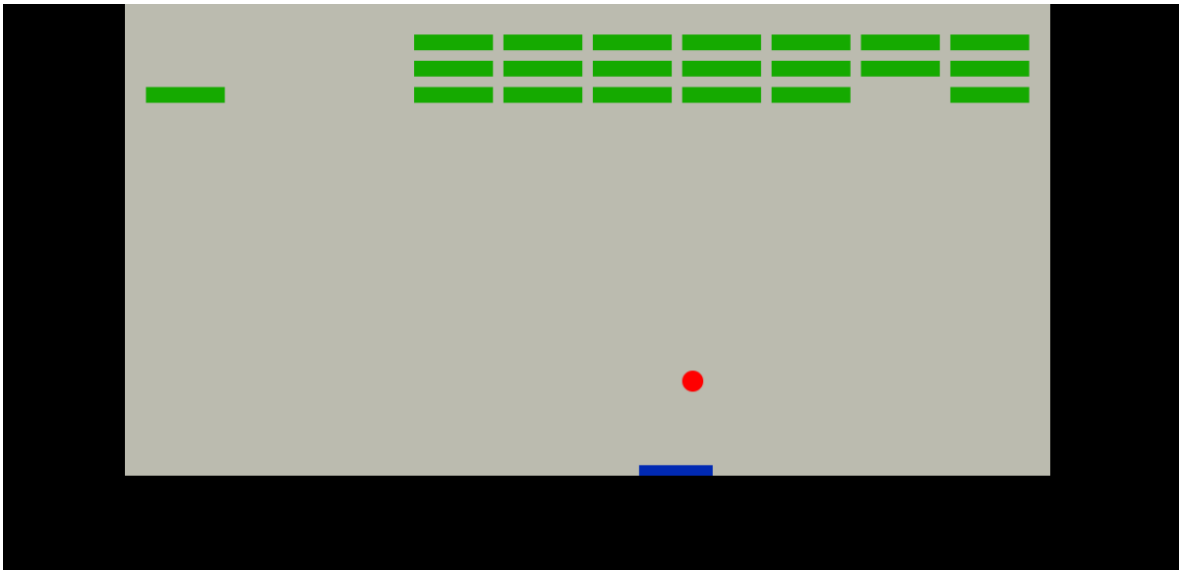
```

```

123.         dx = -dx;
124.     }
125.     if(y + dy < radioBola) {
126.         dy = -dy;
127.     }
128.     else if(y + dy > canvas.height-radioBola) {
129.         if(x > paletaPosX && x < paletaPosX + anchur
aPaleta) {
130.             dy = -dy;
131.         }
132.         else {
133.             clearInterval(juego);
134.             alert("PERDISTE");
135.             document.location.reload();
136.         }
137.     }
138.
139.     if(flechaDerechaPresionada && paletaPosX < canva
s.width-anchuraPaleta) {
140.         paletaPosX += 7;
141.     }
142.     else if(flechaIzquierdaPresionada && paletaPosX
> 0) {
143.         paletaPosX -= 7;
144.     }
145.
146.     x += dx;
147.     y += dy;
148. }
149.
150.     var juego = setInterval(dibujar, 40);
151. </script>
152.
153. </body>
154. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 7. Ejecución del sexto programa.

En la figura 7 podemos observar cómo algunos ladrillos se desaparecieron luego de ser golpeados por la bola.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 9 FASE 7: CONTAR PUNTOS Y GANAR

---

En esta parte del programa se realiza la variable para darle algún valor cuando la bola golpee algún ladrillo y se convierta en un punto y se sumen estos puntos hasta ganar el juego con el máximo de puntos que se puedan obtener

Se crea una variable llamada puntaje la cual controla la cantidad de ladrillos que han sido golpeados por la bola, cada que la bola impacta un ladrillo se le agrega un valor a esta variable hasta que el puntaje es igual al número de ladrillos haciendo que el juego se gane.

En este programa cambiamos el ancho de la paleta para comprobar que el juego se ejecute correctamente y cuando se destruya cada ladrillo suma un punto.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.
6.     <title>JuegoP7</title>
7.
8.     <style>* { padding: 0; margin: 0; } canvas { background:
9.         #BBBBAF; display: block; margin: 0 auto; }</style>
10.
11.     <body style="background-color: #000000;">
12.
13.         <canvas id="miCanvas" width="880" height="450"></canvas>
14.
15.         <script>
16.             var canvas = document.getElementById("miCanvas");
17.             var ctx = canvas.getContext("2d");
18.
19.             var radioBola = 10;
20.             var x = canvas.width/2;
21.             var y = canvas.height-30;
22.             var dx = 10;
23.             var dy = -10;
24.
25.             var alturaPaleta = 10;
26.             var anchuraPaleta = 880;
27.             var paletaPosX = (canvas.width-anchuraPaleta)/2;
28.
29.             var flechaDerechaPresionada = false;
30.             var flechaIzquierdaPresionada = false;
31.
32.             var nroFilasLadrillos = 10;
```

```

33.     var nroColumnasLadrillos = 3;
34.     var anchuraLadrillos = 75;
35.     var alturaLadrillos = 15;
36.     var rellenoLadrillos = 10;
37.     var vacioSuperiorLadrillo = 30;
38.     var vacioIzquierdoLadrillo = 20;
39.
40.     var puntaje = 0;
41.
42.     var ladrillos = [];
43.     for(var c=0; c<nroColumnasLadrillos; c++) {
44.         ladrillos[c] = [];
45.         for(var f=0; f<nroFilasLadrillos; f++) {
46.             ladrillos[c][f] = { x: 0, y: 0, estado: 1 };
47.         }
48.     }
49.
50.     document.addEventListener("keydown", manejadorTeclaP
resionada, false);
51.     document.addEventListener("keyup", manejadorTeclaLib
erada, false);
52.
53.     function manejadorTeclaPresionada(e) {
54.         if(e.keyCode == 39) {
55.             flechaDerechaPresionada = true;
56.         }
57.         else if(e.keyCode == 37) {
58.             flechaIzquierdaPresionada = true;
59.         }
60.     }
61.     function manejadorTeclaLiberada(e) {
62.         if(e.keyCode == 39) {
63.             flechaDerechaPresionada = false;
64.         }
65.         else if(e.keyCode == 37) {
66.             flechaIzquierdaPresionada = false;
67.         }
68.     }
69.     function detectarColision() {
70.         for(var c=0; c<nroColumnasLadrillos; c++) {
71.             for(var f=0; f<nroFilasLadrillos; f++) {
72.                 var b = ladrillos[c][f];
73.                 if(b.estado == 1) {
74.                     if(x > b.x && x < b.x+anchuraLadrill
os && y > b.y && y < b.y+alturaLadrillos) {
75.                         dy = -dy;
76.                         b.estado = 0;
77.                         puntaje++;

```

```

78.                                     if(puntaje == nroFilasLadrillos*
    nroColumnasLadrillos) {
79.                                     alert("GANASTE");
80.                                     document.location.reload();
81.                                     }
82.
83.                                 }
84.                            }
85.                       }
86.                   }
87.           }
88.
89.       function dibujarBola() {
90.           ctx.beginPath();
91.           ctx.arc(x, y, radioBola, 0, Math.PI*2);
92.           ctx.fillStyle = "#FF0000";
93.           ctx.fill();
94.           ctx.closePath();
95.       }
96.
97.       function dibujarPaleta() {
98.           ctx.beginPath();
99.           ctx.rect(paletaPosX, canvas.height-
    alturaPaleta, anchuraPaleta, alturaPaleta);
100.          ctx.fillStyle = "#0029B3";
101.          ctx.fill();
102.          ctx.closePath();
103.      }
104.
105.      function dibujarLadrillos() {
106.          for(var c=0; c<nroColumnasLadrillos; c++) {
107.              for(var r=0; r<nroFilasLadrillos; r++) {
108.                  if(ladrillos[c][r].estado == 1) {
109.                      var posXLadrillo = (r*(anchuraLadril
    los+rellenoLadrillos))+vacioIzquierdoLadrillo;
110.                      var posYLadrillo = (c*(alturaLadrill
    os+rellenoLadrillos))+vacioSuperiorLadrillo;
111.                      ladrillos[c][r].x = posXLadrillo;
112.                      ladrillos[c][r].y = posYLadrillo;
113.                      ctx.beginPath();
114.                      ctx.rect(posXLadrillo, posYLadrillo,
    anchuraLadrillos, alturaLadrillos);
115.                      ctx.fillStyle = "#15AA00";
116.                      ctx.fill();
117.                      ctx.closePath();
118.                  }
119.              }
120.          }
121.      }

```

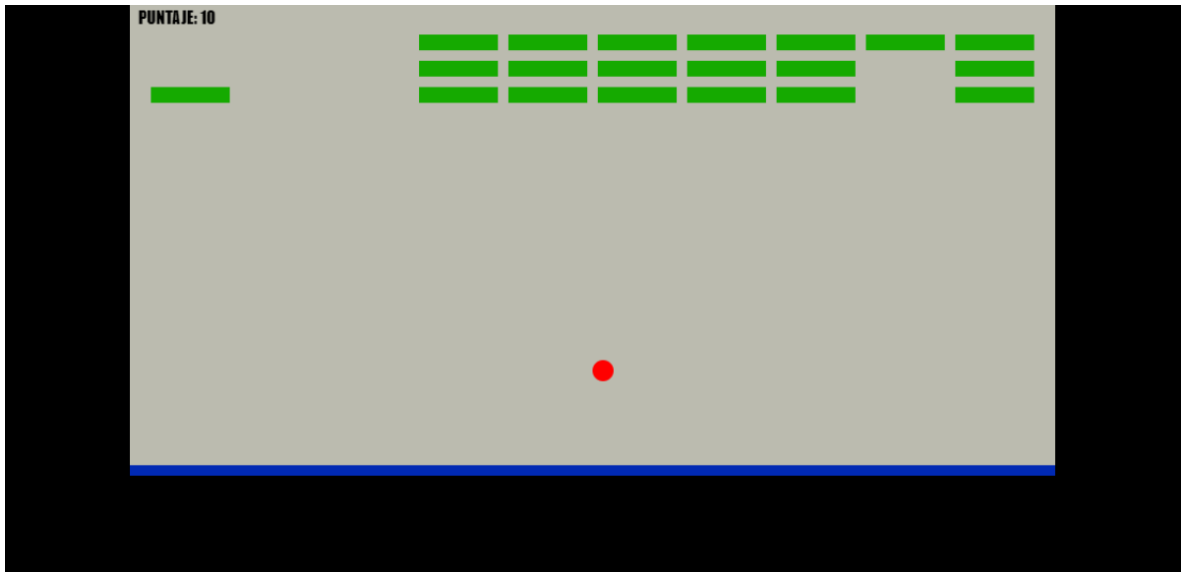
```

122.
123.     function dibujarPuntaje() {
124.         ctx.font = "16px Fantasy";
125.         ctx.fillStyle = "#000000";
126.         ctx.fillText("PUNTAJE: "+puntaje, 8, 20);
127.     }
128.
129.     function dibujar() {
130.         ctx.clearRect(0, 0, canvas.width, canvas.height)
131.         ;
132.         dibujarLadrillos();
133.         dibujarBola();
134.         dibujarPaleta();
135.         dibujarPuntaje();
136.         detectarColision();
137.         if(x + dx > canvas.width-
138.         radioBola || x + dx < radioBola) {
139.             dx = -dx;
140.         }
141.         if(y + dy < radioBola) {
142.             dy = -dy;
143.         }
144.         else if(y + dy > canvas.height-radioBola) {
145.             if(x > paletaPosX && x < paletaPosX + anchur
146.             aPaleta) {
147.                 dy = -dy;
148.             }
149.             else {
150.                 clearInterval(juego);
151.                 alert("PERDISTE");
152.                 document.location.reload();
153.             }
154.         }
155.         if(flechaDerechaPresionada && paletaPosX < canva
156.         s.width-anchuraPaleta) {
157.             paletaPosX += 7;
158.         }
159.         else if(flechaIzquierdaPresionada && paletaPosX
160.         > 0) {
161.             paletaPosX -= 7;
162.         }
163.     }
164.
165.     var juego = setInterval(dibujar, 40);

```

```
166.     </script>
167.
168.     </body>
169.     </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 8. Ejecución del séptimo programa.

En la figura 8 se puede observar como la bola al impactar en los ladrillos estos desaparecen y el puntaje incrementa hasta desaparecer todos los ladrillos y ganar el juego.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.



## 10 FASE 8: CONTROLANDO EL RATÓN

---

En esta parte del programa haremos que la paleta en lugar de ser movida por las flechas sea movida por el mouse.

Esto se obtiene creando una función llamada `function manejadorRaton(e)` a la cual se le da una variable y una condición que al cumplirla hace que la paleta pueda ser desplazada mediante el ratón.

```
1. DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.
6.     <title>JuegoP8</title>
7.
8.     <style>* { padding: 0; margin: 0; } canvas { bac
    kground: #BBBBAF; display: block; margin: 0 auto; }<
    /style>
9. </head>
10.
11.     <body style="background-color: #000000;">
12.
13.     <canvas
    id="miCanvas" width="880" height="450"></canvas>
14.
15.     <script>
16.
17.         var canvas = document.getElementById("miCan
    vas");
18.         var ctx = canvas.getContext("2d");
19.
20.         var radioBola = 10;
21.         var x = canvas.width/2;
22.         var y = canvas.height-30;
23.         var dx = 10;
24.         var dy = -10;
25.
26.         var alturaPaleta = 10;
27.         var anchuraPaleta = 70;
```

```

28.         var paletaPosX = (canvas.width-
           anchuraPaleta)/2;
29.
30.         var flechaDerechaPresionada = false;
31.         var flechaIzquierdaPresionada = false;
32.
33.         var nroFilasLadrillos = 10;
34.         var nroColumnasLadrillos = 3;
35.         var anchuraLadrillo = 75;
36.         var alturaLadrillo = 15;
37.         var rellenoLadrillo = 10;
38.         var vacioSuperiorLadrillo = 30;
39.         var vacioIzquierdoLadrillo = 20;
40.
41.         var puntaje = 0;
42.
43.         var ladrillos = [];
44.         for(var c=0; c<nroColumnasLadrillos; c++) {
45.             ladrillos[c] = [];
46.             for(var f=0; f<nroFilasLadrillos; f++)
47.                 ladrillos[c][f] = { x: 0, y: 0, estado: 1 };
48.             }
49.         }
50.
51.         document.addEventListener("keydown", manejadorTeclaPresionada, false);
52.         document.addEventListener("keyup", manejadorTeclaLiberada, false);
53.
54.         document.addEventListener("mousemove", manejadorRaton, false);
55.
56.         function manejadorTeclaPresionada(e) {
57.             if(e.keyCode == 39) {
58.                 flechaDerechaPresionada = true;
59.             }
60.             else if(e.keyCode == 37) {
61.                 flechaIzquierdaPresionada = true;
62.             }

```

```

63.     }
64.
65.     function manejadorTeclaLiberada(e) {
66.         if(e.keyCode == 39) {
67.             flechaDerechaPresionada = false;
68.         }
69.         else if(e.keyCode == 37) {
70.             flechaIzquierdaPresionada = false;
71.         }
72.     }
73.
74.     function manejadorRaton(e) {
75.         var posXRatonDentroDeCanvas = e.clientX
76.         - canvas.offsetLeft;
77.         if(posXRatonDentroDeCanvas > 0 && posXR
78.         atonDentroDeCanvas < canvas.width) {
79.             paletaPosX = posXRatonDentroDeCanva
80.             s - anchuraPaleta/2;
81.         }
82.         function detectarColision() {
83.             for(var c=0; c<nroColumnasLadrillos; c+
84.             +) {
85.                 for(var r=0; r<nroFilasLadrillos; r
86.                 ++){
87.                     var b = ladrillos[c][r];
88.                     if(b.estado == 1) {
89.                         if(x > b.x && x < b.x+anchu
90.                         raLadrillo && y > b.y && y < b.y+alturaLadrillo) {
91.                             dy = -dy;
92.                             b.estado = 0;
93.                             puntaje++;
94.                             if(puntaje == nroFilasL
95.                             adrillos*nroColumnasLadrillos) {
96.                                 alert("GANASTE");
97.                                 document.location.r
98.                                 eload();
99.                             }
100.                        }
101.                    }
102.                }
103.            }
104.        }
105.    }

```

```

96.         }
97.     }
98. }
99.
100.    function dibujarBola() {
101.        ctx.beginPath();
102.        ctx.arc(x, y, radioBola, 0, Math.PI*2);
103.        ctx.fillStyle = "#FF0000";
104.        ctx.fill();
105.        ctx.closePath();
106.    }
107.    function dibujarPaleta() {
108.        ctx.beginPath();
109.        ctx.rect(paletaPosX, canvas.height-
    alturaPaleta, anchuraPaleta, alturaPaleta);
110.        ctx.fillStyle = "#0029B3";
111.        ctx.fill();
112.        ctx.closePath();
113.    }
114.    function dibujarLadrillos() {
115.        for(var c=0; c<nroColumnasLadrillos; c+
    +) {
116.            for(var r=0; r<nroFilasLadrillos; r
    ++){
117.                if(ladrillos[c][r].estado == 1)
    {
118.                    var brickX = (r*(anchuraLad
    rillo+rellenoLadrillo))+vacioIzquierdoLadrillo;
119.                    var brickY = (c*(alturaLadr
    illo+rellenoLadrillo))+vacioSuperiorLadrillo;
120.                    ladrillos[c][r].x = brickX;
121.                    ladrillos[c][r].y = brickY;
122.                    ctx.beginPath();
123.                    ctx.rect(brickX, brickY, an
    churaLadrillo, alturaLadrillo);
124.                    ctx.fillStyle = "#15AA00";
125.                    ctx.fill();
126.                    ctx.closePath();
127.                }
128.            }
129.        }

```

```

130.     }
131.     function dibujarPuntaje() {
132.         ctx.font = "16px Fantasy";
133.         ctx.fillStyle = "#000000";
134.         ctx.fillText("PUNTAJE:
    "+puntaje, 8, 20);
135.     }
136.
137.     function dibujar() {
138.         ctx.clearRect(0, 0, canvas.width, canva
    s.height);
139.         dibujarLadrillos();
140.         dibujarBola();
141.         dibujarPaleta();
142.         dibujarPuntaje();
143.         detectarColision();
144.
145.         if(x + dx > canvas.width-
    radioBola || x + dx < radioBola) {
146.             dx = -dx;
147.         }
148.         if(y + dy < radioBola) {
149.             dy = -dy;
150.         }
151.         else if(y + dy > canvas.height-
    radioBola) {
152.             if(x > paletaPosX && x < paletaPosX
    + anchuraPaleta) {
153.                 dy = -dy;
154.             }
155.             else {
156.                 clearInterval(juego);
157.                 alert("PERDISTE");
158.                 document.location.reload();
159.             }
160.         }
161.
162.         if(flechaDerechaPresionada && paletaPos
    X < canvas.width-anchuraPaleta) {
163.             paletaPosX += 7;
164.         }

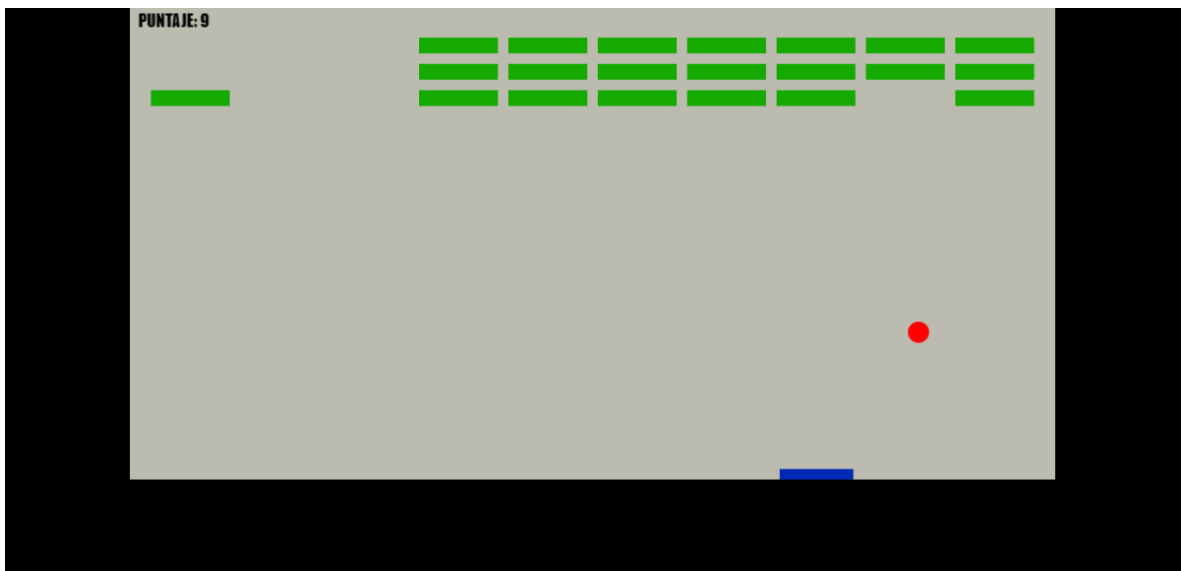
```

```

165.         else if(flechaIzquierdaPresionada && pa
    letaPosX > 0) {
166.             paletaPosX -= 7;
167.         }
168.
169.         x += dx;
170.         y += dy;
171.     }
172.
173.     var juego = setInterval(dibujar, 40);
174. </script>
175.
176. </body>
177. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 9. Ejecución del octavo programa.

En la figura 9 se puede observar como la paleta es controlada de derecha a izquierda mediante el uso del ratón.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 11 FASE 9: FINALIZANDO EL JUEGO

---

En esta parte del programa ya se agregan los toques finales tales como vidas para el jugador, ocultar el mouse entre otras.

Se crea una variable `var vidas = 3` con la instrucción de controlar las vidas que tiene dentro del juego cada participante y se crea otra variable `canvas.style.cursor = 'none'` para ocultar el mouse dentro del campo del juego, también se crea la instrucción `vidas--`; la cual lleva la cuenta de las vidas que tiene y que ha perdido.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.
6.     <title>Juego completo</title>
7.
8.     <style>* { padding: 0; margin: 0; } canvas { background:
    #BBBBAF; display: block; margin: 0 auto; * {cursor: none;} }
    </style>
9.
10.    </head>
11.
12.    <body style="background-color: #000000;">
13.
14.    <canvas id="miCanvas" width="880" height="450"></canvas>
15.
16.    <script>
17.        var canvas = document.getElementById("miCanvas");
18.        var ctx = canvas.getContext("2d");
19.
20.        var bolaRadio = 10;
21.        var x = canvas.width/2;
22.        var y = canvas.height-30;
23.        var dx = 10;
24.        var dy = -10;
25.
26.        var alturaPaleta = 10;
27.        var anchuraPaleta = 70;
28.        var paletaPosX = (canvas.width-anchuraPaleta)/2;
29.
30.        var flechaDerechaPresionada = false;
31.        var flechaIzquierdaPresionada = false;
32.
33.        var nroFilasLadrillos = 10;
34.        var nroColumnasLadrillos = 3;
```

```

35.     var anchuraLadrillo = 75;
36.     var alturaLadrillo = 15;
37.     var rellenoLadrillo = 10;
38.     var vacioSuperiorLadrillo = 30;
39.     var vacioIzquierdoLadrillo = 20;
40.
41.     var puntaje = 0;
42.
43.     var vidas = 3;
44.
45.     var colorFigura = "#ff0000";
46.     var colorBola = "#ff0000";
47.     var colorPaleta = "#0029B3";
48.     var colorLadrillo = "#15AA00";
49.     var colorTexto = "#000000";
50.
51.     canvas.style.cursor = 'none';
52.
53.     var ladrillos = [];
54.     for(var c=0; c<nroColumnasLadrillos; c++) {
55.         ladrillos[c] = [];
56.         for(var f=0; f<nroFilasLadrillos; f++) {
57.             ladrillos[c][f] = { x: 0, y: 0, estado: 1 };
58.         }
59.     }
60.
61.     document.addEventListener("keydown", manejadorTeclaP
resionada, false);
62.     document.addEventListener("keyup", manejadorTeclaLib
erada, false);
63.     document.addEventListener("mousemove", manejadorRato
n, false);
64.
65.     function manejadorTeclaPresionada(e) {
66.         if(e.keyCode == 39) {
67.             flechaDerechaPresionada = true;
68.         }
69.         else if(e.keyCode == 37) {
70.             flechaIzquierdaPresionada = true;
71.         }
72.     }
73.
74.     function manejadorTeclaLiberada(e) {
75.         if(e.keyCode == 39) {
76.             flechaDerechaPresionada = false;
77.         }
78.         else if(e.keyCode == 37) {
79.             flechaIzquierdaPresionada = false;
80.         }

```



```

81.     }
82.
83.     function manejadorRaton(e) {
84.         var posXRatonDentroDeCanvas = e.clientX -
            canvas.offsetLeft;
85.         if(posXRatonDentroDeCanvas > 0 && posXRatonDentr
            oDeCanvas < canvas.width) {
86.             paletaPosX = posXRatonDentroDeCanvas -
                anchuraPaleta/2;
87.         }
88.     }
89.
90.     function detectarColision() {
91.         for(var c=0; c<nroColumnasLadrillos; c++) {
92.             for(var f=0; f<nroFilasLadrillos; f++) {
93.                 var b = ladrillos[c][f];
94.                 if(b.estado == 1) {
95.                     if(x > b.x && x < b.x+anchuraLadrill
                        o && y > b.y && y < b.y+alturaLadrillo) {
96.                         dy = -dy;
97.                         b.estado = 0;
98.                         puntaje++;
99.                         if(puntaje == nroFilasLadrillos*
                            nroColumnasLadrillos) {
100.                             alert("GANASTE");
101.                             document.location.reload();
102.                         }
103.                     }
104.                 }
105.             }
106.         }
107.     }
108.
109.     function dibujarBola() {
110.         ctx.beginPath();
111.         ctx.arc(x, y, bolaRadio, 0, Math.PI*2);
112.         // SE UTILIZA EL COLOR PREVIAMENTE DEFINIDO
113.         ctx.fillStyle = colorBola;
114.         ctx.fill();
115.         ctx.closePath();
116.     }
117.     function dibujarPaleta() {
118.         ctx.beginPath();
119.         ctx.rect(paletaPosX, canvas.height-
            alturaPaleta, anchuraPaleta, alturaPaleta);
120.         ctx.fillStyle = colorPaleta;
121.         ctx.fill();
122.         ctx.closePath();
123.     }

```

```

124.         function dibujarLadrillos() {
125.             for(var c=0; c<nroColumnasLadrillos; c++) {
126.                 for(var f=0; f<nroFilasLadrillos; f++) {
127.                     if(ladrillos[c][f].estado == 1) {
128.                         var ladrilloX = (f*(anchuraLadrillo+
rellenoLadrillo))+vacioIzquierdoLadrillo;
129.                         var ladrilloY = (c*(alturaLadrillo+r
ellenoLadrillo))+vacioSuperiorLadrillo;
130.                         ladrillos[c][f].x = ladrilloX;
131.                         ladrillos[c][f].y = ladrilloY;
132.                         ctx.beginPath();
133.                         ctx.rect(ladrilloX, ladrilloY, anchu
raLadrillo, alturaLadrillo);
134.                         ctx.fillStyle = colorLadrillo;
135.                         ctx.fill();
136.                         ctx.closePath();
137.                     }
138.                 }
139.             }
140.         }
141.
142.         function dibujarPuntaje() {
143.             ctx.font = "16px Fantasy";
144.             ctx.fillStyle = colorTexto;
145.             ctx.fillText("PUNTAJE: "+puntaje, 8, 20);
146.         }
147.
148.         function dibujarVidas() {
149.             ctx.font = "16px Fantasy";
150.             ctx.fillStyle = colorTexto;
151.             // SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES
152.             ctx.fillText("VIDAS: "+vidas, canvas.width-
65, 20);
153.         }
154.
155.         function dibujar() {
156.             ctx.clearRect(0, 0, canvas.width, canvas.height)
;
157.             dibujarLadrillos();
158.             dibujarBola();
159.             dibujarPaleta();
160.             dibujarPuntaje();
161.             dibujarVidas();
162.             detectarColision();
163.
164.             if(x + dx > canvas.width-
bolaRadio || x + dx < bolaRadio) {
165.                 dx = -dx;
166.             }

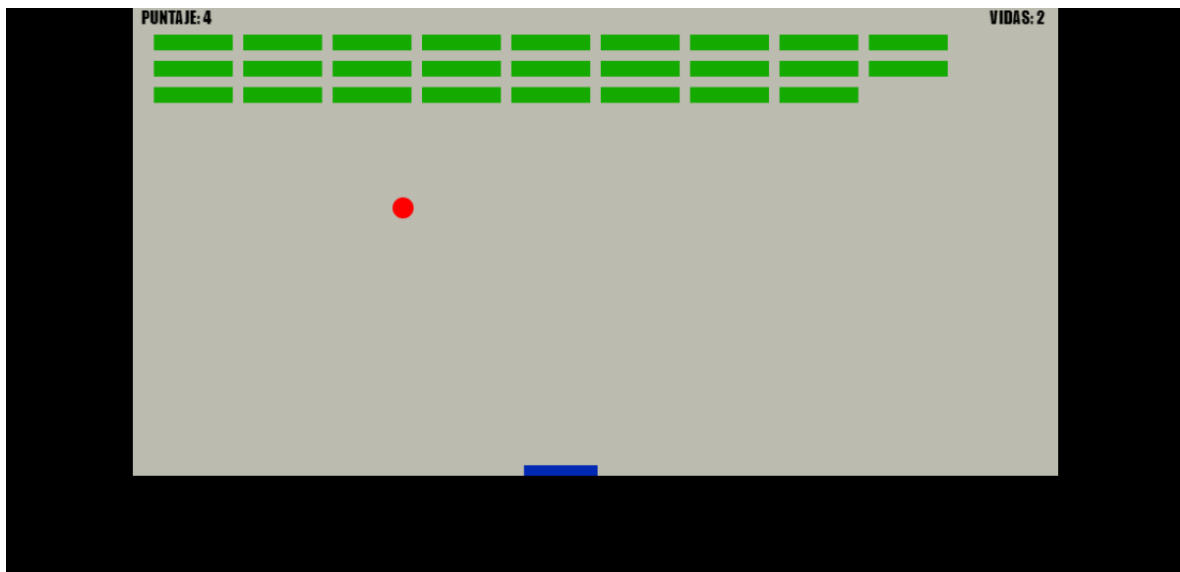
```

```

167.         if(y + dy < bolaRadio) {
168.             dy = -dy;
169.         }
170.         else if(y + dy > canvas.height-bolaRadio) {
171.             if(x > paletaPosX && x < paletaPosX + anchur
aPaleta) {
172.                 dy = -dy;
173.             }
174.             else {
175.                 vidas--;
176.                 if(!vidas) {
177.                     alert("PERDISTE");
178.                     document.location.reload();
179.                 }
180.                 else {
181.                     x = canvas.width/2;
182.                     y = canvas.height-30;
183.                     dx = 3;
184.                     dy = -3;
185.                     paletaPosX = (canvas.width-
anchuraPaleta)/2;
186.                 }
187.             }
188.         }
189.
190.         if(flechaDerechaPresionada && paletaPosX < canva
s.width-anchuraPaleta) {
191.             paletaPosX += 7;
192.         }
193.         else if(flechaIzquierdaPresionada && paletaPosX
> 0) {
194.             paletaPosX -= 7;
195.         }
196.
197.         x += dx;
198.         y += dy;
199.
200.         requestAnimationFrame(dibujar);
201.     }
202.
203.     dibujar();
204. </script>
205.
206. </body>
207. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 10. Ejecución del juego totalmente completado.

En la imagen 10 podemos observar el juego ya completado totalmente, y en el podemos observar las vidas y el puntaje que lleva el jugador durante el juego y la desaparición del mando dentro del canvas.



## 12 CONCLUSIONES

---

En conclusión, podemos visualizar la creación de un juego en 2D utilizando 9 programas diferentes, cada uno más avanzado que el anterior para llegar al objetivo de construir este juego.

Este juego fue realizado utilizando html y Javascript en su totalidad, desarrollamos el juego gracias a la consecución de 9 pasos, los cuales dentro de cada uno aprendimos las funciones y comandos para en un futuro ejecutarlos en un nuevo proyecto en el mundo de la programación.



## 13 BIBLIOGRAFÍA

---

[https://developer.mozilla.org/es/docs/Games/Workflows/Famoso\\_juego\\_2D\\_usando\\_JavaScript\\_puro/Construye\\_grupo\\_bloques](https://developer.mozilla.org/es/docs/Games/Workflows/Famoso_juego_2D_usando_JavaScript_puro/Construye_grupo_bloques)