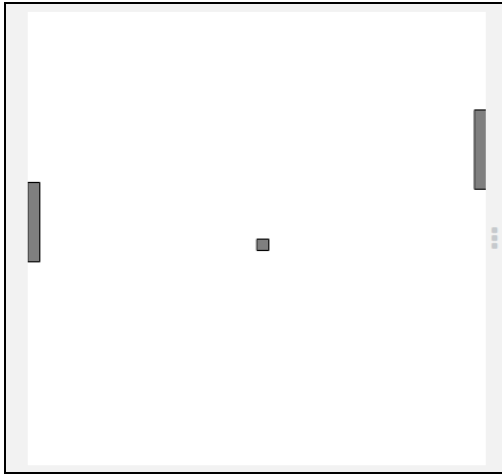


Tema		Programación Orientada a Objetos	
Descripción de Clase		Esta clase ayuda a los estudiantes a pensar en programación orientada a objetos. Los estudiantes diseñan una Pelota y una Barra en clase y crean objetos utilizando la clase para usarlo en su programa.	
Clase		PRO-C1	
Tiempo de clase		45 mins	
Objetivos		<ul style="list-style-type: none"> <li>• Diseñar una clase de Paddle (barra) y Pelota (Ball).</li> <li>• Crear objetos usando las clases Paddle y Ball y usarlas en el programa</li> <li>• Guardar objetos en variables.</li> </ul>	
Recursos Requeridos		<ul style="list-style-type: none"> <li>• Recursos para el Profesor               <ul style="list-style-type: none"> <li>○ Computadora portátil con acceso a internet</li> <li>○ Auriculares con micrófono</li> <li>○ Cuaderno y pluma</li> <li>○ Smartphone</li> </ul> </li> <li>• Recursos para el Estudiante               <ul style="list-style-type: none"> <li>○ Computadora portátil con acceso a internet</li> <li>○ Auriculares con micrófono</li> <li>○ Cuaderno y pluma</li> </ul> </li> </ul>	
Estructura de la Clase		<b>Rompiendo el hielo</b> <b>Actividad dirigida por el Profesor</b> <b>Actividad dirigida por el Estudiante</b> <b>Conclusión</b>	<b>5 mins</b> <b>15 min</b> <b>20 min</b> <b>5 min</b>
<b>CONTEXTO</b> <ul style="list-style-type: none"> <li>• Nuevo estilo de programación usando el estilo orientado a objetos</li> </ul>			
Pasos de la Clase	Acciones del profesor		Acciones del Estudiante
<b>Paso 1:</b> <b>Rompiendo el Hielo</b> <b>5 mins</b>	<i>Felicite al estudiante por sus esfuerzos durante la clase anterior</i>  <i>"¡Hola! &lt;nombre del estudiante&gt;! Me da gusto verte de nuevo ¿Cómo estuvo tu día hoy?</i>		<i>El estudiante le cuenta al profesor sobre su día.</i>

	"La última clase te fue muy bien. Espero verte trabajar con el mismo esfuerzo en la clase de hoy".	
	<p><i>Repase el progreso que se hizo en clase.</i></p> <p>¿Te acuerdas de lo que hicimos la última clase?</p>	<p><b>Anime al Estudiante a Responder (AER):</b> ¡Sí! Estuvimos trabajando en el juego de Pong. Hicimos dos barras y la pelota. Hicimos que una barra se moviera con el ratón.</p>
	<p><i>Felicite al estudiante por lo que haya recordado efectivamente.</i></p> <p>Perfecto. Se ve que recuerdas el juego en el que estuvimos trabajando. Esto significa que has pensado en el juego, lo que está muy bien. Un buen programador siempre piensa en lo que está desarrollando.</p>	<p><i>El estudiante escucha.</i></p>
	<p><i>Siente la base para el tema de esta clase.</i></p> <p>En la clase de hoy vamos a aprender conceptos importantes de programación. Vamos a aprender sobre un estilo de programación que los programadores expertos utilizan al escribir códigos.</p> <p>La clase de hoy es la más difícil de todo el curso. Si logramos superarla, nos vamos a divertir mucho en las siguientes clases.</p> <p>¿Listo para el reto?</p>	<p><i>El estudiante escucha.</i></p> <p><b>AER: ¡Sí!</b></p>

	Con esto en mente, iniciemos la clase de hoy. Voy a compartir mi pantalla contigo.	
<b>El Profesor Comienza Compartir Pantalla</b>		
<p style="text-align: center;"><b><u>RETO</u></b></p> <ul style="list-style-type: none"> <li>● Presentar el estilo de programación orientado a objetos</li> <li>● Muestre cómo diseñar una clase Paddle y cómo crear objetos usando la clase.</li> <li>● Rételos a crear su propio objeto de paddle.</li> </ul>		
<b>Paso 2:</b> <b>Actividad</b> <b>dirigida por el</b> <b>Profesor</b> <b>15 min</b>	<p><i>Muestre al estudiante el código de la clase anterior.</i></p> <p><i>Revise si el estudiante recuerda las funciones de cada parte del código.</i></p> <p>Hablemos del código de la clase pasada.</p> <p>¿Recuerdas estos?</p> <p>Haga clic en el <a href="#">Enlace de la Actividad del Profesor 1</a> y haga clic en Archivo&gt; <b>Duplicar</b></p>	<p><b>AER</b> ¡Sí! .</p>
<b>Código</b> <div> <pre> 1 function setup() { 2   createCanvas(400, 400); 3 } 4 5 function draw() { 6   background("white"); 7   rect(390,mouseY,10,70); 8   rect(0,160,10,70); 9   rect(185,185,10,10) 10 } </pre> </div>		
<b>Resultado:</b>		

		
	<p>¿Recuerdas para qué sirve la primera instrucción rect()?</p> <p>¿Para qué sirven los números de las instrucciones rect()?</p> <p>¿Para qué sirve la segunda instrucción rect()?</p> <p>¿Para qué sirve la tercera instrucción rect()?</p>	<p><b>AER</b></p> <p>Para trazar el player paddle.</p> <p>Los números dentro indicaban la posición de x, y así como el alto y el ancho.</p> <p>Para trazar el computer paddle.</p> <p>Para trazar la pelota.</p>
<p><b>Código</b></p> <pre> 1 function setup() { 2   createCanvas(400, 400); 3 } 4 5 function draw() { 6   background("white"); 7   rect(390,mouseY,10,70); 8   rect(0,160,10,70); 9   rect(185,185,10,10) 10 }</pre>		
	<p><i>Hable sobre el código difícil de leer. Introduzca los comentarios en el código para explicar qué hace el código.</i></p> <p>Estas instrucciones que damos a la</p>	

	<p>computadora van trazando lo que queremos en la pantalla, pero, ¿estas instrucciones te explican lo que están haciendo?</p> <p>El código dice que está trazando un rectángulo, pero no te dice si es para el paddle o ball. Alguien que esté viendo el código tal vez no sepa lo que el código busca hacer.</p> <p>Nuestro programa, como lo tenemos en este momento, es muy difícil de leer. Un buen programa debe decir al lector lo que está haciendo. Una de las formas para hacer a nuestro programa más amigable con los lectores es añadir comentarios.</p> <p>Añadimos comentarios usando <code>"/"</code> (dos diagonales).</p> <p>Al comentar en el código le podemos decir a los lectores de nuestro programa lo que nuestro código busca hacer. La computadora ignora estas líneas.</p> <p>Agreguemos comentarios a nuestro código.</p> <p><i>El profesor pide al estudiante qué comentarios deben ponerse en diferentes lugares del programa.</i></p>	<p><b>AER</b> No.</p> <p><i>El estudiante da ideas de comentarios.</i></p>
--	--	--

## Código

Comentarios añadidos a la pantalla:

```

> sketch.js
1 function draw() {
2   //limpia la pantalla
3   background("white");
4   //dibuja la Raqueta del Jugador
5   rect(390,World,mouseY,10,70);
6   //dibuja la Raqueta de la Computadora
7   rect(0,150,10,70);
8   //dibuja la pelota
9   rect(200,200,10,10);
10 }
  
```

	<p><i>Enfóquese en cómo los comentarios hacen que su código sea legible. Raqueta se usará como Paddle.</i></p> <p>¿Piensas que esto hace que el código sea más fácil de leer? ¿Crees que otros entenderán lo que estás tratando de hacer con este programa?</p> <p><b>DEJE DE COMPARTIR PANTALLA PARA VER LA CARA DEL ALUMNO</b></p> <p>Los programadores siempre añaden comentarios para que otros programadores entiendan su código.</p>	<p><b>AER</b></p> <p>¡Sí! .</p>
	<p>Los comentarios no son suficientes. Leer el programa debería decirnos lo que hace.</p> <p><i>Presente el estilo de programación orientado a objetos</i></p> <p>Los comentarios que añadimos al programa lo hicieron fácil de comprender. Pero el programa en sí no nos dice lo que está haciendo.</p>	

	<p>Un buen programa debe explicar lo que hace por sí mismo, aun sin los comentarios. De hecho, un buen programa debería leerse como leer una historia. Debes poder leer cualquier parte del código y saber lo que pasa en el código.</p> <p>De hecho vamos a aprender este estilo para escribir programas.</p> <p>¿Puedes pensar por qué y en dónde es útil este estilo de programación?</p> <p>Este tipo de programas fáciles de leer son muy útiles cuando varios programadores trabajan juntos en un solo proyecto, como un juego importante o una aplicación. Como estos programas se pueden leer con facilidad, nos ayudan a desarrollar una lógica compleja en el código pero de forma sencilla.</p> <p>¿No crees que sería emocionante aprender el tipo de programación que se usa para desarrollar aplicaciones complejas?</p>	<p><b>AER:</b> [variado]</p>
	<p><i>Conecta las propiedades y funciones de los objetos del mundo real con las propiedades y funciones de los objetos en programación.</i></p> <p>En este estilo de programación escribimos el código como si todo en el mundo de la programación fuera un objeto, justo como en el mundo real.</p>	<p><b>AER:</b> ¡Sí!</p>

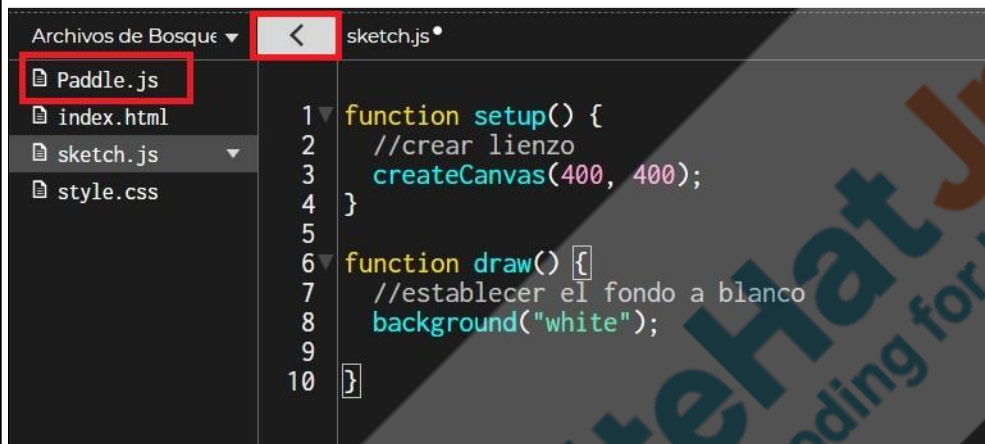
	<p>En el mundo real todo a nuestro alrededor es un objeto. Cada objeto tiene ciertas propiedades y funciones.</p> <p>Las <b>PROPIEDADES</b> son las cualidades/características del objeto. Las <b>FUNCIONES</b> son lo que el objeto puede 'hacer'.</p> <p><b>El profesor explica las definiciones con su smartphone en la mano.</b> "Pensemos en un smartphone. ¿Qué propiedades tiene un smartphone?  ¿Qué funciones tiene?</p> <p>Todos los objetos del mundo real tienen PROPIEDADES y FUNCIONES.</p> <p>Pensemos en otro objeto, una pluma. ¿Qué propiedades tiene una pluma?  ¿Qué funciones tiene una pluma?</p> <p>Al igual que con los objetos del mundo real, los programadores ven todo en su mundo de programación como hecho de objetos reales que tienen propiedades y funciones.</p>	<p><b>AER</b> Tamaño, color, grosor, etc.</p> <p><b>AER</b> Llamar a alguien, tomar fotografías, etc.</p> <p><b>AER:</b> color, tamaño, etc.</p> <p><b>AER:</b> escribir</p>
	<p><i>Conecte la fabricación de los objetos en la vida real a cómo los programadores crean objetos.</i> <i>Pida al estudiante que le hable sobre los objetos de nuestro juego.</i></p> <p>¿Cuáles son los objetos de nuestro juego?</p>	<p><b>AER</b> Dos barras y una pelota</p>



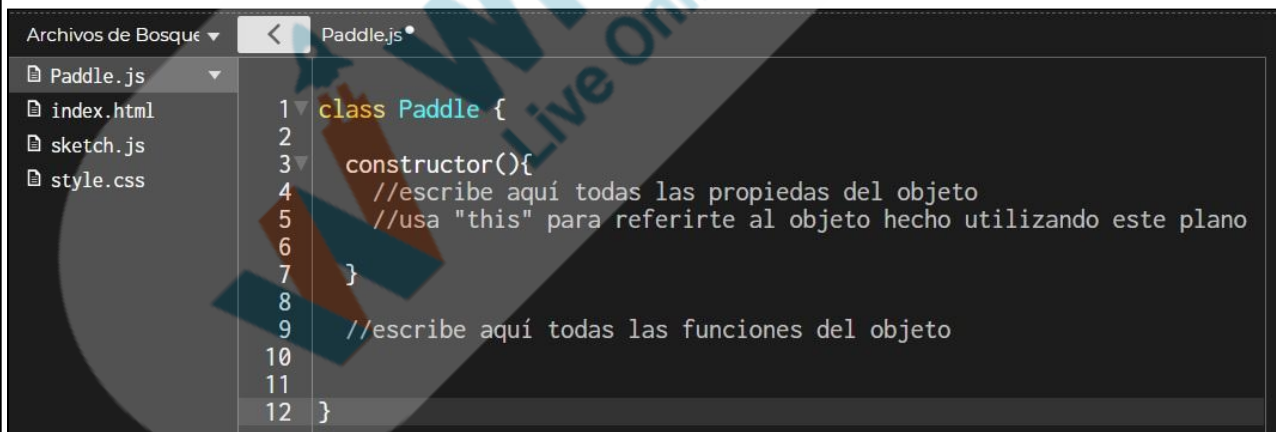
	<p>¡Exacto! Aprendamos cómo crear estos objetos en nuestro programa.</p> <p>Antes de ello, ¿te puedes imaginar cómo se hacen objetos nuevos en una fábrica?</p>	<p><b>AER</b></p> <p>Primero se hace un molde/plano que tiene el diseño del objeto. Los objetos nuevos se hacen con este diseño en mente.</p>
	<p>Los programadores usan la misma técnica cuando crean objetos en su programa.</p> <p>Un programador que va a crear el objeto de una barra primero hace el diseño y le asigna las propiedades y funciones del objeto. Con base en este diseño, el programador creará tantos objetos de barra (Paddle) como quiera en el juego.</p> <p>Este diseño se llama CLASE en programación. Esta forma de ver el mundo de la programación en objetos se llama <b>programación orientada a objetos</b>.</p> <p>Déjame mostrarte cómo se ve una clase muestra.</p>	<p><i>El estudiante escucha.</i></p>
	<p><i>Muestre el archivo de clase paddle al estudiante.</i></p> <p><i>Muestre las áreas en las que se añaden las propiedades y funciones.</i></p> <p>El programador crea una clase de esta forma.</p> <p>Así es como creamos la Clase Paddle.</p>	<p><i>El estudiante escucha, observa y aprende.</i></p>

--	--	--

**Abre la carpeta de archivos al hacer clic en la flecha izquierda.**



**Abre el archivo Paddle.js**



	<p>Dentro de la Clase Paddle o el diseño de Paddle, podemos definir todas las propiedades y funciones de nuestra barra.</p> <p>¿Cuáles son las propiedades para esta barra?</p>	<p><b>AER:</b> Ancho, Alto, Posición en x y Posición en y.</p>
--	---	--

	<p>Asignemos todas las propiedades de la barra a la clase/el diseño de nuestra barra.</p> <p>Cuando se hace un objeto que use este diseño, se invoca o se llama a lo que está en el constructor ().</p> <p>Vamos a asignar todas las propiedades de la barra en el constructor ().</p> <p><i>El profesor asigna valores a this.width, this.height, this.xPosition, this.yPosition. Ahora dirá al estudiante que "this" se refiere al objeto que se creará usando este diseño/clase de Paddle.</i></p> <p>Se pueden cambiar el ancho, alto, la posición en x y la posición en y del objeto más adelante.</p>	<p><i>El estudiante escucha, observa y aprende.</i></p>
--	---	---

## Código

```

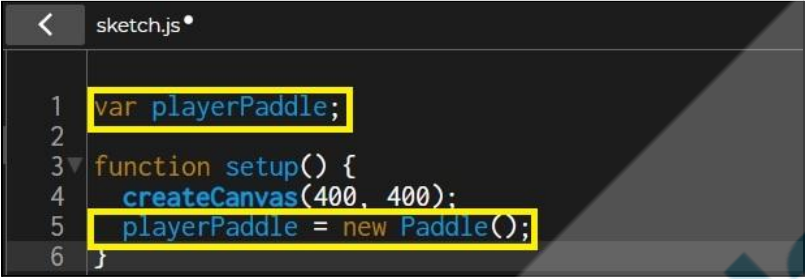
< Paddle.js
1 class Paddle {
2   //se usa el constructor para inicializar un objeto
3   constructor(){
4     this.xPosition = 0;
5     this.yPosition = 0;
6     this.width = 10;
7     this.height = 70;
8   }
9 }

```

	<p>Necesitamos indicarle a la computadora dónde encontrar la Clase Paddle.</p> <p>Esto se hace haciendo un pequeño cambio en el archivo index.html</p> <p>Index.html es el primer archivo que se carga al ejecutar tu aplicación. Este contiene la información de qué archivos e información deben cargarse. Todos</p>	
--	--	--

	<p>los archivos de javascript (el lenguaje en el que estamos escribiendo el código) se cargan usando script tags.</p> <p>Un script tag comienza así &lt;script&gt; y termina así &lt;/script&gt;.</p> <p>"src" o la fuente (source) que está dentro del script tag indica qué archivo cargar.</p> <p>Muestre cómo se agrega el archivo Paddle.js a los script tags del index.html.</p>	
<pre> 3  &lt;head&gt; 4    &lt;script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.1.9/p5.js"&gt; &lt;/script&gt; 5    &lt;script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.1.9/addons/ p5.sound.min.js"&gt;&lt;/script&gt; 6    &lt;link rel="stylesheet" type="text/css" href="style.css"&gt; 7    &lt;meta charset="utf-8" /&gt; 8    &lt;script src="Paddle.js"&gt;&lt;/script&gt; 9  &lt;/head&gt; 10 &lt;body&gt; 11   &lt;script src="sketch.js"&gt;&lt;/script&gt; 12 &lt;/body&gt; 13 &lt;/html&gt; </pre>		
	<p><i>Crea un objeto PlayerPaddle usando la clase Paddle (raqueta).</i></p> <p><i>Introduzca las variables como espacios de memoria en los que las computadoras almacenan objetos.</i></p> <p>Parece que nuestra clase paddle está lista. Eliminemos nuestra antigua instrucción rect() para PlayerPaddle y vamos a crear una raqueta nueva basada en la clase/el diseño de Paddle que definimos.</p> <p><i>El profesor elimina la antigua</i></p>	<p><i>El estudiante escucha, observa y aprende.</i></p>

	<p><i>instrucción rect() para el player paddle. Ejecute el código para mostrar que en el resultado no se muestra el player paddle aún.</i></p> <p>Podemos crear un nuevo Paddle dando esta instrucción a la computadora:</p> <p><b>new Paddle();</b></p> <p>Recuerda que usamos el punto y coma para indicarle a la computadora que este es el fin de la línea. Es igual al punto final en el español.</p> <p>Hemos creado el objeto Paddle. Ahora la computadora debe guardar el objeto en algún lado.</p> <p>¿Dónde crees que la computadora almacenará el objeto?</p> <p>La computadora almacena cada objeto en su memoria. Vamos a dar un nombre a la memoria usando algo que se llaman variables y almacenando los objetos en esta.</p> <p>Las variables retienen los objetos en la memoria de la computadora. Podemos lograr esto dando las siguientes instrucciones a la computadora:</p> <p><b>var playerPaddle;</b> <b>playerPaddle = new Paddle();</b></p> <p>En esta instrucción creamos una variable que puede almacenar un objeto. Le dimos el nombre playerPaddle. Ahora vamos a guardar el objeto Paddle en esta variable.</p>	<p><b>AER</b></p> <p>¿En la memoria?</p>
--	---	--

	<p>Los nombres de las variables no deben tener espacios y deben comenzar con una letra, "_" o con "\$".</p> <p>Los nombres de las variables también deben ser significativos, es decir, deben decirnos lo que alojan.</p>	
<b>Código</b>		
		
	<p><i>Cambia las propiedades xPosition y yPosition para el objeto PlayerPaddle</i></p> <p>Ahora, podemos asignar o cambiar las propiedades de este objeto playerPaddle.</p> <p>El alto y ancho del PlayerPaddle ya están definidos en el diseño. Cambiemos la xPosition y la yPosition del playerPaddle.</p> <p>¿Cuál debería de ser la xPosition y la yPosition del playerPaddle?</p>	<p><b>AER</b></p> <pre>playerPaddle.xPosition = 390; playerPaddle.yPosition = World.mouseY;</pre>
<b>Código</b>		

```

< sketch.js
1  var playerPaddle;
2
3  function setup() {
4    createCanvas(400, 400);
5    playerPaddle = new Paddle();
6  }
7
8  function draw() {
9    background("white");
10   playerPaddle.xPosition=390;
11   playerPaddle.yPosition=mouseY;
12   rect(0,165,10,70);
13   rect(200,200,10,10);
14 }

```

Parece que tenemos nuestro objeto playerPaddle listo. Ejecutemos el código y veamos si se traza el playerPaddle en nuestra pantalla.

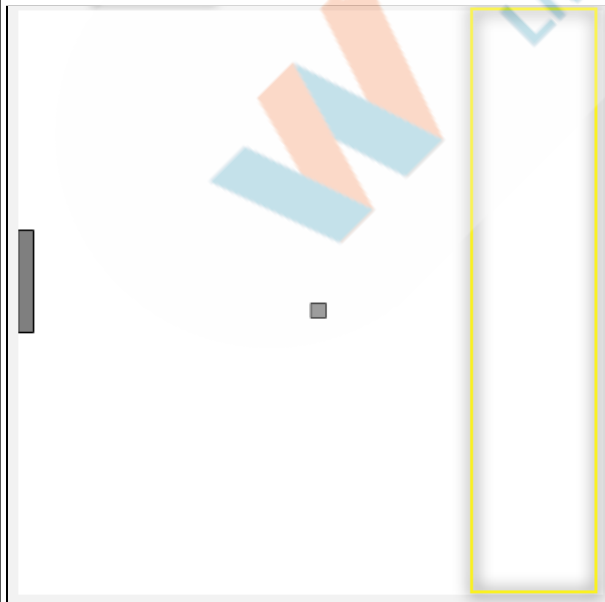
¿Qué piensas que pasará si ejecutemos el código ahora?

*Ejecuta el código.*

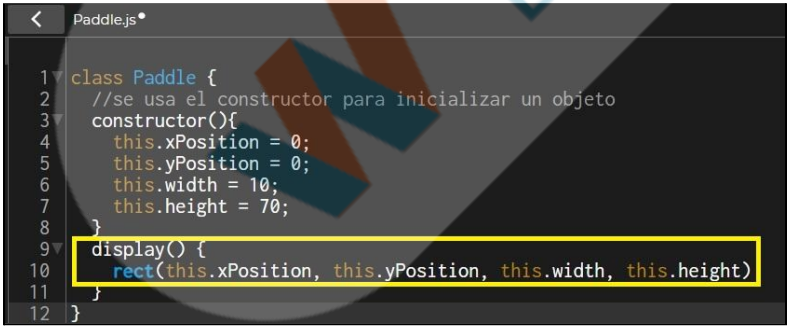
*El estudiante predice cuál será el resultado.*

*El estudiante ve el resultado.*

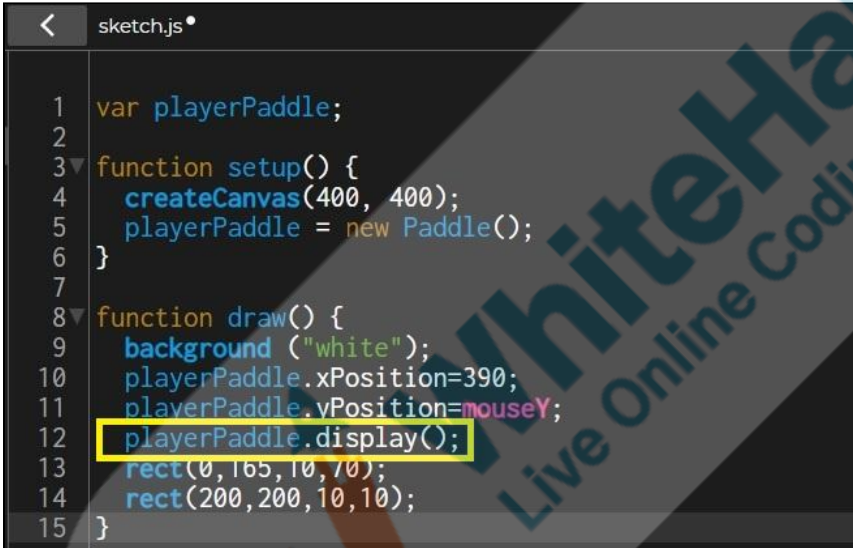
**Resultado después de ejecutar:** <No Player Paddle>





	<p>No pasó nada ¿Por qué?</p>	<p><i>El estudiante pausa para pensar por qué no sucede nada en la pantalla. El estudiante le dirá las razones por las que esto pudo haber pasado.</i></p>
	<p><i>Escribe mostrar función para el objeto playerPaddle para mostrar la barra.</i></p> <p>Esto es porque aunque nuestro objeto playerPaddle se creó con todas las propiedades, no sabe cómo hacer nada. No sabe cómo mostrarse a sí mismo Necesitamos definir en nuestra clase/diseño Paddle cómo debe mostrarse el paddle. Hagamos eso.</p> <p><i>El profesor añade esta línea a la clase Paddle.</i></p> <pre>display() {   rect(this.xPosition,this.yPosition,this.wid   th,this.height) ;</pre>	<p><i>El estudiante observa y aprende.</i></p>
<b>Código</b> 		
	<p>Digamos al playerPaddle cómo mostrarse y ejecutemos el código para ver si el playerPaddle se muestra.</p> <p>¿Tienes una idea de cómo hacerlo?</p>	<p><b>AER</b> playerPaddle.display</p>

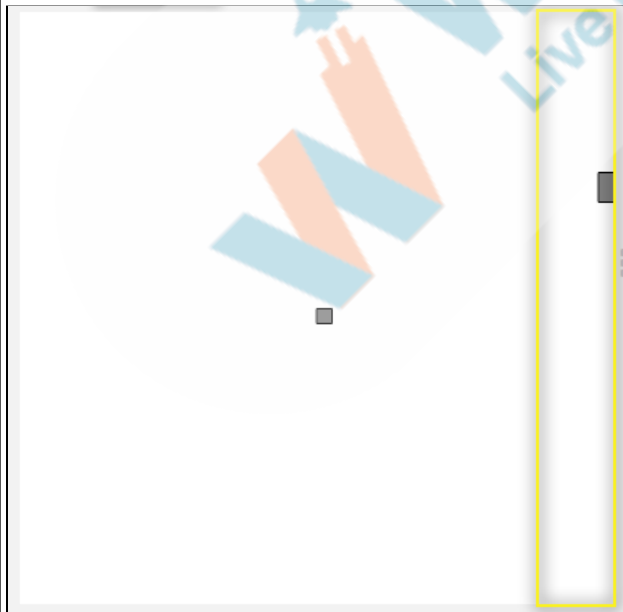


	<p><i>El profesor agrega playerPaddle.display() dentro de draw().</i></p> <p>Cuando le pedimos a la computadora que haga algo por nosotros, le añadimos los corchetes. En el mundo de programación a esto se le llaman las funciones.</p> <p>Ejecuta el código. El playerPaddle debe mostrarse en la pantalla.</p>	
<b>Código</b>		
		
	<p><i>Indique cómo trabajar correctamente con los objetos relacionándolo con la vida real.</i></p> <p><i>Muéstrele cómo otras propiedades del objeto paddle se pueden modificar.</i></p> <p>¡Wow! Ya lo logramos. Tenemos el PlayerPaddle en la pantalla de nuevo.</p> <p>Vamos a recordar rápido lo que hemos hecho hasta ahora:</p>	<p><b>AER</b></p> <p>Resumen</p> <ul style="list-style-type: none"><li>• Escribir comentarios para hacer el programa fácil de leer.</li><li>• Crear un plano/diseño del objeto y asignar propiedades y funciones a los objetos.</li><li>• Crear nuevos objetos y guardarlos en la</li></ul>

	<p>Como programador vas a ser el DIOS del mundo que creas. Puedes crear cualquier objeto en este mundo del juego y puedes cambiar sus propiedades y destruirlo.</p> <p>Piensa en qué sería esto en el mundo real.</p> <p>Si tuvieras el objeto de una botella (bottle) podrías cambiar sus propiedades diciendo <code>bottle.color = "red"</code> y la botella se volvería roja.</p> <p>O podrías configurar el tamaño de la pelota asignándole valor a <code>bottle.size</code></p> <p>También puedes decir <code>bottle.fill()</code> y la botella se llenaría.</p> <p>¿No es genial?! Esto es lo que puede hacer un programador en el mundo de la programación. Pueden asignar valores nuevos para los objetos en su mundo y cambiar los objetos por completo.</p> <p>Podríamos haber asignado valores diferentes al objeto <code>playerPaddle</code> y se habría mostrado diferente dependiendo de esos valores.</p>	<p>memoria de la computadora.</p> <ul style="list-style-type: none"> <li>● Cambiar las propiedades de los objetos.</li> </ul> <p><i>El estudiante menciona cambiar el ancho y el alto del objeto <code>PlayerPaddle</code>.</i></p>
<b>Código</b>		

```
sketch.js
1  var playerPaddle;
2
3  function setup() {
4    createCanvas(400, 400);
5    playerPaddle = new Paddle();
6  }
7
8  function draw() {
9    //limpia la pantalla
10   background("white");
11   //dibuja el Player Paddle
12   playerPaddle.xPosition=390;
13   playerPaddle.vPosition=mouseY;
14   playerPaddle.height=20;
15   playerPaddle.display();
16
17   //dibuja el Computer Paddle
18
19   //dibuja la Pelota
20   rect(200,200,10,10);
```

**Resultado:**



	<p>Como un reto ¿por qué no creas otro Paddle Object para el computerPaddle?</p> <p>¿Qué tendrías que hacer para crear un nuevo objeto computerPaddle?</p>	<p><b>AER</b></p> <ul style="list-style-type: none"> <li>- Invocar nuevo Paddle().</li> <li>- Guardarlo en una variable llamada computerPaddle().</li> <li>- Configura la xPosition y la yPosition para mostrarlo.</li> <li>- Invoca al computerPaddle() para que se muestre.</li> </ul>
	<p>¿Por qué no compartes tu pantalla y tratas de crear el objeto computerPaddle?</p>	<p><i>El estudiante comparte su pantalla.</i></p> <p><i>Abre la Actividad del Estudiante y reajusta el código y añade más código a esta.</i></p>
<b>El Profesor Detiene Compartir Pantalla</b>		
	<p>Ahora es tu turno. Por favor comparte tu pantalla conmigo.</p>	
<ul style="list-style-type: none"> <li>• <b>Pida al estudiante que presione la tecla ESC para volver al Panel</b></li> <li>• <b>Guíe al Estudiante para que comience a Compartir Pantalla</b></li> <li>• <b>El profesor deberá entrar en Pantalla Completa</b></li> </ul>		
<p style="text-align: center;"><u><b>ACTIVIDAD</b></u></p> <ul style="list-style-type: none"> <li>• <b>El estudiante crea su ComputerPaddle usando la clase Paddle.</b></li> <li>• <b>El estudiante asigna diferentes propiedades al objeto ComputerPaddle.</b></li> </ul>		
<p><b>Paso 3:</b></p> <p><b>Actividad dirigida por el Estudiante</b></p> <p><b>20 min</b></p>	<p><i>Guíe al estudiante para que cree un paddle object con sus propias propiedades usando la clase Paddle.</i></p> <p>Creemos un nuevo objeto Paddle.</p> <p><i>Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.</i></p>	<p><i>El estudiante abre la <a href="#">Actividad del estudiante 1</a> y hace clic en Archivo&gt;Duplicar</i></p> <p><i>El estudiante escribe el código sobre la etapa actual del juego para crear un objeto Paddle nuevo.</i></p>

## Código

```

< sketch.js
1  var playerPaddle,computerPaddle;
2
3  function setup() {
4      createCanvas(400, 400);
5      playerPaddle = new Paddle();
6      new Paddle();
7  }
8
9  function draw() {
10     //limpia la pantalla
11     background ("white");
12     //dibuja el Player Paddle
13     playerPaddle.xPosition=390;
14     playerPaddle.yPosition=mouseY;
15     playerPaddle.height=20;

```

Vamos a guardar el nuevo Objeto Paddle en una variable y darle el nombre computerPaddle.

*Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.*

*El estudiante escribe el código para crear un objeto ComputerPaddle.*

## Código

```

< sketch.js
1  var playerPaddle computerPaddle;
2
3  function setup() {
4    createCanvas(400, 400);
5    playerPaddle = new Paddle();
6    computerPaddle = new Paddle();
7  }
8
9  function draw() {
10   //limpia la pantalla
11   background("white");
12   //dibuja el Player Paddle
13   playerPaddle.xPosition=390;
14   playerPaddle.yPosition=mouseY;
15   playerPaddle.height=20;

```

Asignemos la xPosition y la yPosition del computerPaddle.

El estudiante asigna la xPosition y la yPosition al computerpaddle.

Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.

## Código

```

< sketch.js Saved: 15 seconds ago
5  playerPaddle = new Paddle();
6  computerPaddle = new Paddle();
7  }
8
9  function draw() {
10   //clear the screen
11   background("white");
12   //draw the Player Paddle
13   playerPaddle.xPosition=390;
14   playerPaddle.yPosition=mouseY;
15   playerPaddle.height=20;
16   playerPaddle.display();
17
18   //draw the Computer Paddle
19   computerPaddle.xPosition=0;
20   computerPaddle.yPosition=150;

```



	<p>Perfecto. Creamos el objeto y le asignamos una posición.</p> <p>Ahora pidamos al objeto que se muestre en la pantalla.</p> <p>Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.</p>	<p><i>El estudiante invoca mostrar el Computer Paddle.</i></p>
--	--	--

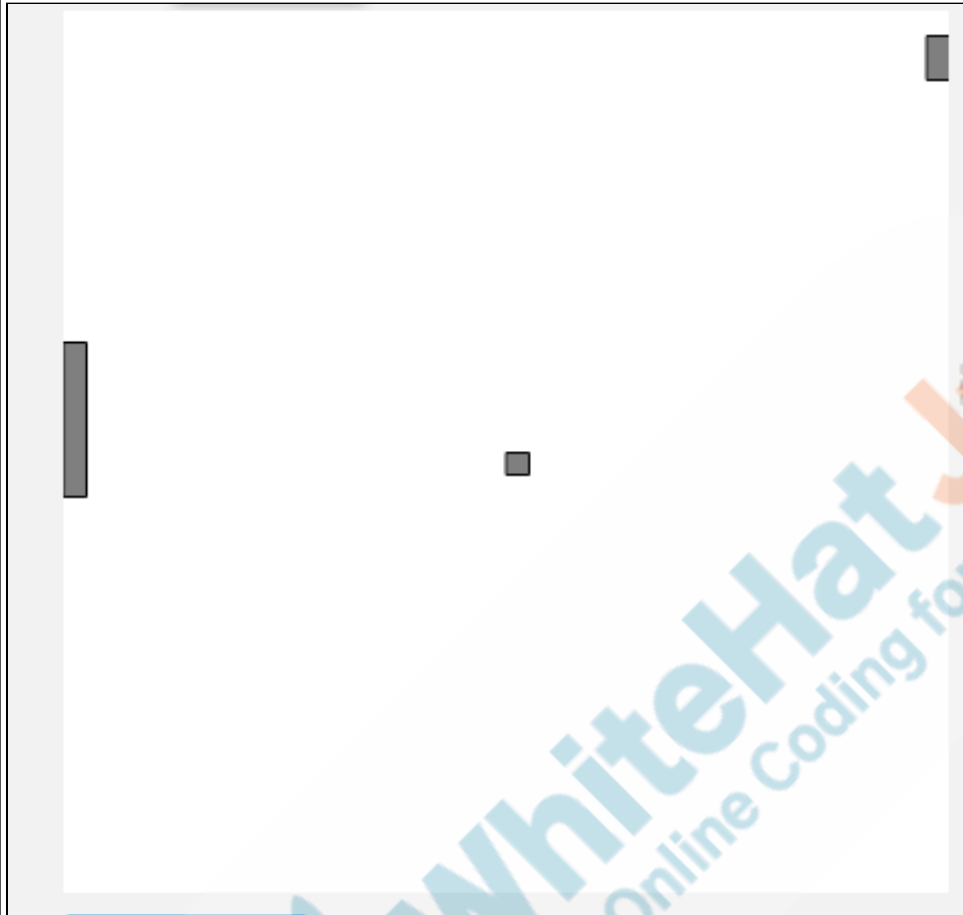
## Código

```

< sketch.js
1  var playerPaddle, computerPaddle;
2
3  function setup() {
4      createCanvas(400, 400);
5      playerPaddle = new Paddle();
6      computerPaddle = new Paddle();
7  }
8
9  function draw() {
10     //limpia la pantalla
11     background("white");
12     //dibuja el Player Paddle
13     playerPaddle.xPosition=390;
14     playerPaddle.yPosition=mouseY;
15     playerPaddle.height=20;
16     playerPaddle.display();
17
18     //dibuja el Computer Paddle
19     computerPaddle.xPosition=0;
20     computerPaddle.yPosition=150;

```

**Resultado:**



¡Maravilloso! Has creado un objeto en este mundo del juego ¡El objeto es todo tuyo! Puedes cambiar cualquiera de sus propiedades ¿Quieres modificar el ancho y el alto del ComputerPaddle?

*Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.*

*Ejecute el código para ver el resultado.*

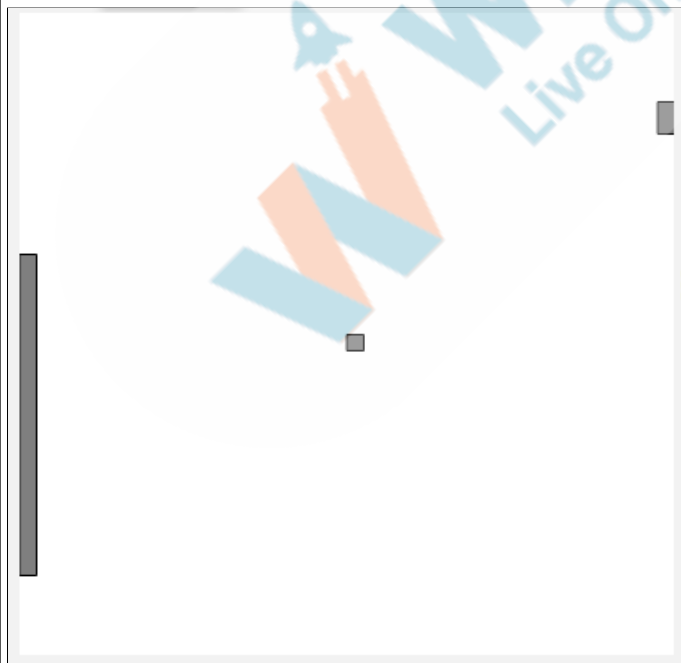
*El estudiante modifica el código para cambiar el ancho y alto del Computer Paddle.*



## Código



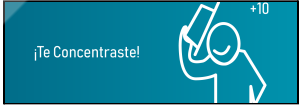
```
sketch.js
4  createCanvas(400, 400);
5  playerPaddle = new Paddle();
6  computerPaddle = new Paddle();
7  }
8
9  function draw() {
10   //limpia la pantalla
11   background("white");
12   //dibuja el Player Paddle
13   playerPaddle.xPosition=390;
14   playerPaddle.yPosition=mouseY;
15   playerPaddle.height=20;
16   playerPaddle.display();
17
18   //dibuja el Computer Paddle
19   computerPaddle.xPosition=0;
20   computerPaddle.yPosition=150;
21
22   computerPaddle.height=200;
23
24   computerPaddle.display();
25   //dibuja la Pelota
```

## Resultado:



	<p>Ahora sabes cómo crear un objeto en el mundo del juego. Ahora ya puedes crear cualquier cosa en este mundo.</p> <p>De hecho, el mundo del juego, en sí, es un objeto. Por esto tenemos propiedades en el mundo del juego como World.x, World.y y así sucesivamente ¡¿No es genial?!</p> <p>¡Todo lo que está en el mundo del juego es un objeto!</p>	<i>El estudiante escucha.</i>
<b>El Profesor Guía al Estudiante para Dejar de Compartir Pantalla</b>		
<p align="center"><b>COMENTARIOS DE RETROALIMENTACIÓN</b></p> <ul style="list-style-type: none"> <li>• <b>Valore y felicite al estudiante por tratar de aprender un concepto difícil.</b></li> <li>• <b>Descubra cómo se siente después de esta sesión.</b></li> <li>• <b>Revise y verifique su comprensión.</b></li> </ul>		
<p><b>Paso 4:</b> <b>Conclusión</b> <b>(5 min)</b></p>	<p><i>Felicite al estudiante por sus esfuerzos.</i> <i>Repase el material de la clase que se abarcó hoy.</i> <i>Haga que el estudiante se sienta emocionado por la siguiente clase.</i></p> <p>Esta forma de escribir programas como si todo fuera un OBJETO se llama PROGRAMACIÓN ORIENTADA A OBJETOS. Este es uno de los conceptos más poderosos en programación.</p> <p>¿Cuál piensas que es la diferencia entre el estilo de programación orientada a objetos y entre el estilo que seguimos en la clase pasada?</p>	<p><b>AER:</b> Esta forma de escribir un programa (orientada a objetivos) es más larga que el estilo anterior, pero se puede leer más fácilmente. Cuando tenemos los diseños/clases listos, es fácil crear nuevos objetos. Quienquiera que lea nuestro</p>

	<p>Digamos que estamos trabajando en un juego espacial que tiene muchas naves espaciales. En este estilo de programación podemos diseñar el plano de la nave espacial una vez y cada que necesitemos una nueva nave solo la invocamos con <b>new Spaceship().ship()</b>.</p> <p>¿No es esta una forma más sencilla de escribir nuestros programas?</p>	<p>programa puede entender lo que sucede en el juego.</p> <p><b>AER:</b> ¡Sí!</p>
	<p>El estilo de programación orientado a objetivos hace nuestro código fácil de leer. Como mencionamos antes, los buenos programas son como las buenas historias. Cualquiera que lea una parte de nuestro programa debe poder saber qué pasa en nuestro programa.</p> <p>¿Piensas que nuestro programa es más fácil de leer ahora? ¿Crees que cualquiera que lea nuestro código entenderá lo que tratamos de hacer?</p>	<p><b>AER:</b> ¡Sí! .</p>
	<p>Hay algunas cosas que aprendimos hoy que son nuevas y que te llevará tiempo acostumbrarte a usarlas, y eso está bien.</p> <p>Tal vez quieras ver el código que escribiste y reflexionar después de la clase.</p> <p>Por ahora ¿podemos resumir rápidamente lo que aprendimos hoy?</p>	<p><b>AER</b> - Añadir comentarios a nuestro programa</p>

		<ul style="list-style-type: none"> <li>- Un nuevo estilo de programación en el que todo es un <b>OBJETO</b> con <b>PROPIEDADES</b> y <b>FUNCIONES</b>.</li> <li>- Crear el diseño de un objeto.</li> <li>- Crear objetos usando el diseño.</li> <li>- Cambiar las propiedades del objeto.</li> </ul>
	<p>¡Te felicito!</p> <p>Muy bien, parece que tuvimos muchos aprendizajes en la clase de hoy. Aunque fue un poco difícil lo hiciste muy bien.</p> <p>En la siguiente clase vamos a crear más objetos y a asignarles propiedades adicionales, por ejemplo haremos que la pelota rebote.</p> <p>¿No es interesante?</p>	<p>Asegúrese de felicitar al estudiante al menos 2 veces por:</p> <div>    </div>
<b>Generalidades del Proyecto</b>	<p><b>OBJETIVOS</b></p> <p><b>Objetivo del Proyecto:</b> Hoy, has aprendido los conceptos básicos relacionados con la Programación Orientada a Objetos (OOP por sus siglas en inglés; Object-Oriented Programming), y has creado una barra y una pelota utilizando esos conceptos. Un objeto en código, representa un objeto físico en el mundo. El objeto físico tiene propiedades, como forma, color, tamaño, etc.</p>	<p><i>El estudiante habla sobre el proyecto con el profesor.</i></p>

	<p>En este proyecto, tendrás que practicar y aplicar lo aprendido en clase, y crear una maqueta de una ciudad, de la cual eres Alcalde.</p> <p><b>Historia:</b> Como alcalde de la ciudad, necesitas crear suficientes viviendas para que no haya personas sin hogar en tu ciudad. Tienes una cantidad fija de espacio para construir viviendas y edificios para cada familia de la ciudad.</p> <p>Me emociona mucho ver la solución que darás a tu proyecto. Sé que te va a ir muy bien.</p> <p>¡Adiós!</p>	
<p style="text-align: center;"><b>El profesor hace Clic en</b> <span style="background-color: red; color: white; padding: 2px 10px; border-radius: 15px; display: inline-block;">✕ Finalizar Clase</span></p>		
<b>Actividades Adicionales</b>	<p>Como un reto ¿por qué no tratas de crear la clase Ball?</p> <p><i>Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.</i></p>	<p><i>El estudiante escribe el código para crear la clase Ball.</i></p> <p><i>Todo el código será similar a la clase Paddle a excepción del nombre de clase que cambiará de Paddle a Ball y los valores del ancho y el alto cambiarán.</i></p>

## Código

```

< Ball.js
1 class Ball
2 {
3   constructor()
4   {
5     this.xPosition=0;
6     this.yPosition=0;
7     this.radius=20;
8   }
9   display()
10  {
11    circle(this.xPosition,this.yPosition,this.radius);
12  }
13 }

```

Muy bien, ya puedes comprar un objeto de pelota (Ball). E incluso puedes crear múltiples objetos de pelota. Imagina un juego de pong con dos pelotas en lugar de una. ¿No sería más difícil?

Creemos un objeto Ball desde la clase Ball. Esto lo haremos de forma similar a como creamos el objeto paddle en la clase Paddle.

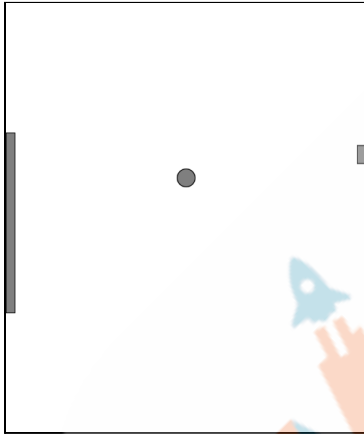
*Observe al estudiante y revise si comete errores ortográficos. Guíe al estudiante para que escriba el código.*

*El estudiante escribe el código para crear al menos uno o más objetos ball en el juego.*

## Código

```
playerPaddle.height=20;  
playerPaddle.display();  
  
//dibuja el Computer Paddle  
computerPaddle.xPosition=0;  
computerPaddle.yPosition=150;  
  
computerPaddle.height=200;  
  
computerPaddle.display();  
//dibuja la Pelota  
ball.xPosition = 200;  
ball.yPosition = 200;  
ball.display();
```

## Resultado:



Actividad	Nombre de la Actividad	Enlaces
Enlace para Actividad del Profesor 1	Pong Etapa 1	<a href="https://editor.p5js.org/whitehatjr/sketches/48OYFKOWa">https://editor.p5js.org/whitehatjr/sketches/48OYFKOWa</a>
Enlace de la Actividad del Profesor 2	Pong Etapa 1.7 (Para referencia del profesor)	<a href="https://editor.p5js.org/whitehatjr/sketches/HLY9ZPTNz">https://editor.p5js.org/whitehatjr/sketches/HLY9ZPTNz</a>
Enlace de la Actividad del Estudiante 1	Pong Etapa 1.5	<a href="https://editor.p5js.org/Abhijeet/sketches/Uq6g3jYl">https://editor.p5js.org/Abhijeet/sketches/Uq6g3jYl</a>