



Tema	PROGRAMA DE COMPUTADORA
Descripción de la clase	<b>Esta clase establece la importancia de aprender a programar en el contexto actual de robots impulsados por Inteligencia Artificial. El alumno comienza a diseñar el juego "romper ladrillos" dibujando y posicionando las figuras de los objetos del juego.</b>
Clase	<b>PRO-T</b>
Duración de la clase	<b>50 mins.</b>
Objetivo	<ul style="list-style-type: none"> <li>● Establecer la filosofía detrás del curso.</li> <li>● Interpretar un programa como un conjunto de instrucciones dadas a una computadora para realizar una tarea en particular.</li> <li>● Comprender cómo las instrucciones dadas a una computadora son diferentes de las instrucciones dadas a un ser humano.</li> <li>● Comenzar a diseñar un juego de ruptura simple. Crear los objetos del juego y animar la paleta del jugador.</li> </ul>
Recursos requeridos	<ul style="list-style-type: none"> <li>● Recursos de la maestra: <ul style="list-style-type: none"> <li>○ Audífonos con micrófono.</li> <li>○ Una botella con agua.</li> <li>○ Libreta y pluma.</li> <li>○ Mesa y silla</li> <li>○ Laptop y su cargador.</li> <li>○ Velocidad de internet: velocidad de descarga <math>\geq 20</math> Mbps.</li> </ul> </li> <li>● Recursos del alumno: <ul style="list-style-type: none"> <li>○ Laptop con conexión a internet.</li> <li>○ Audífonos con micrófono.</li> <li>○ Libreta y pluma.</li> <li>○ Velocidad de internet: velocidad de descarga <math>\geq 20</math> Mbps.</li> </ul> </li> <li>● Factores de higiene: <ul style="list-style-type: none"> <li>○ Usa un fondo limpio y claro.</li> <li>○ Mantén la cámara encendida siempre.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Asegúrate de que haya suficiente iluminación en tu cara (la luz debe estar sobre tu cara y no sobre tu espalda).</li> <li>○ Mantén tu teléfono en silencio (consulta la sección de cómo realizar una clase de prueba exitosa para obtener más detalles).</li> </ul>	
<b>Estructura de la clase</b>	<b>Rompiendo el hielo</b> <b>Actividad dirigida por la maestra 1</b> <b>Actividad dirigida por el alumno 1</b> <b>Conclusión</b>	<b>10 minutos</b> <b>15 minutos</b> <b>15 minutos</b> <b>10 minutos</b>
<b>ROMPIENDO EL HIELO - 10 minutos</b>		
<b><u>CONTEXTO</u></b>		
	<ul style="list-style-type: none"> <li>● Establecer una conexión con el alumno.</li> <li>● Interpretar un programa como un conjunto de instrucciones enviadas a una computadora, para realizar una tarea en particular.</li> <li>● Comprender en qué se diferencian las instrucciones enviadas a una computadora, de las instrucciones enviadas a un ser humano.</li> </ul>	
Acción de la maestra	Acción del alumno	
<p>¡Hola, &lt;nombre del alumno&gt; ¡Soy &lt;nombre del maestra&gt;!</p> <p>Voy a ser tu maestra en este programa. Conozcámonos un poco antes de comenzar.</p> <p><b>Nota:</b> anima al alumno a compartir sobre sí mismo, sus intereses, etc.</p> <p><b>También comparte un poco sobre ti.</b></p> <p>Primero, compartiré una breve introducción sobre mí. Me encantaría saber más sobre ti.</p> <p>He realizado &lt;certificación XYZ&gt; y mis pasatiempos son &lt;áreas de interés del maestra&gt;.</p> <p>&lt;Nombre del alumno&gt;, por favor dime en qué grado estás y qué áreas te interesan.</p>		

*Espera su respuesta.*

Además, ¿puedes contarme tus expectativas de esta clase?

*El alumno comparte sobre sí mismo, sobre sus intereses, etc.*

*El alumno comparte sobre sus expectativas de la clase.*

### **Establece la agenda.**

*Pídele al alumno que invite a los padres a unirse a la clase. En caso de que el parent no esté presente, o si está ocupado, pídele que se una en los últimos 10 minutos si es posible.*

Permíteme decirte rápidamente **lo que haremos en esta clase de prueba.**

- Te enseñaré los conceptos básicos de JavaScript, el cual es un lenguaje de programación muy popular, que se puede usar para diseñar sitios web, apps móviles y juegos.
- Comenzarás a crear una app del juego de romper ladrillos para un solo jugador, usando JavaScript.
- Hacia el final, presentaré la estructura del curso y los resultados de los alumnos, que te ayudarán a comprender exactamente lo que ofrecemos.

*El alumno invita al parent a unirse a la clase.*

Entonces, <nombre del alumno>, ¿por qué quieres aprender a programar?

*Los alumnos comparten sus razones.*

*Explora todas las razones por las que el alumno quiere aprender a programar. Algunas cosas de las que el alumno podría hablar son la robótica, la creación de juegos, la creación de software y aplicaciones, etc.*

*Aprecia las razones dadas por el alumno usando palabras como "increíble", "impresionante", "genial", etc.*

También me gustaría agregar una razón por la cual alguien querría aprender a programar hoy.

¿Has oido hablar de la "**inteligencia artificial**" y de cómo las computadoras se están volviendo tan inteligentes como los humanos?

Hoy en día, en los lugares de trabajo, estamos principalmente con otros humanos más que con computadoras. Sin embargo, 10 o 15 años después, podríamos estar trabajando a diario con computadoras y robots a nuestro alrededor, más que con humanos.

¿Puedes decirme algunos ejemplos de la vida real en los que se pueden usar robots para ayudar a los humanos?

- Los robots se pueden usar para construir casas o hacer un camino sobre las montañas.
- Los robots pueden ayudar a los pacientes, transportándolos en el hospital.
- Los robots pueden ayudarnos a enseñar.
- Los robots pueden ayudarnos a proteger nuestro país, incluso nuestras propias casas.
- Los robots se pueden utilizar para evitar delitos.

Y, por supuesto, nos gustaría vivir en un mundo donde todos puedan comprar un robot que también pueda hacer nuestro trabajo doméstico.

Bueno, ahora puedes ver cómo las computadoras y los robots cambiarán la forma en que vivimos en el futuro, por lo que debemos saber cómo dar instrucciones a la computadora para que haga las cosas.

Dar instrucciones a la computadora es muy diferente de dar instrucciones a un humano.

**REA:** variada.

Permíteme darte **una demostración** para ayudarte a entender cómo dar instrucciones a una computadora es diferente de dar instrucciones a un humano.

*Coloca una botella con agua a tu lado para que sea visible para el alumno a través de la cámara web.*

Soy un humano. Dame instrucciones para beber agua de esta botella.

*Sigue las sencillas instrucciones dadas por el alumno para beber agua de la botella.*

Ahora, imagina que soy un robot. Ahora dame instrucciones para beber agua de la botella.

*Sigue todas las instrucciones literalmente.*

*Haz que la situación sea graciosa para el alumno.*



**REA:** beba agua de esta botella.

**REA:** algunos ejemplos:

- El alumno puede pedirte que bebas agua de la botella. Intentarás beber agua sin abrir la botella.
- El alumno puede pedirte que abras la botella. Intentarás destapar la botella sin girar primero la tapa.
- El alumno podría pedirte que gires la tapa de la botella. Intentarás girar la tapa en el sentido de las agujas del reloj, lo que la apretará aún más.

**P:** ¿cuál crees que fue la diferencia entre dar instrucciones a un humano y dar instrucciones a un robot/computadora?

*Anima al alumno a descifrar la diferencia entre las instrucciones dadas a los humanos y las instrucciones dadas a los robots.*

R: un robot necesitaba instrucciones más precisas/exactas para completar una tarea. Un humano podría entendernos sin las instrucciones exactas.

Si resumimos, a grandes rasgos, los pasos para pedirle a un robot que beba agua de una botella pueden seguir este orden.

- Identificar la botella sobre la mesa.
- Recoger la botella.
- Girar la tapa en sentido contrario a las agujas del reloj, hasta que la tapa se afloje.
- Levantar la tapa y colocarla sobre la mesa.
- Acercar la botella a su boca e inclinarla para beber el agua.

¿Viste cómo luchaste para dar **instrucciones exactas** al robot?

Bueno, al dar instrucciones a un robot/computadora, debemos tener en cuenta tres cosas.

- Primero, las **instrucciones deben ser exactas**, lo que significa que debemos decir exactamente lo que queremos que haga la computadora.
- En segundo lugar, las **instrucciones deben seguir una secuencia adecuada**. Debemos saber qué paso realizar primero y qué paso a continuación. Al igual que antes de comenzar a beber el agua de la botella, debemos abrir la tapa de la botella.
- Tercero, las **instrucciones deben estar en un lenguaje que la computadora pueda entender**.

*El alumno trata de diferenciar entre las instrucciones dadas a los humanos y las instrucciones dadas a los robots.*

REA: si.

Entendemos los dos primeros puntos con la demostración.

Ahora, para comprender el tercer punto, ¿me pueden decir si las computadoras pueden entender español?

En todo el mundo, los humanos hablan diferentes idiomas como español (en España), francés (en Francia), hindi (en India), mandarín (en China).

Usamos estos lenguajes para comunicarnos entre nosotros. Usando estos lenguajes, podemos hacer que otro humano entienda lo que queremos decirle.

Ahora, ambos podemos hablar <nombre del idioma nativo común que hablan el alumno y la maestra>.

Pero si queremos hablar con personas chinas (o de cualquier otro país que no sea el país del alumno y de la maestra), debemos aprender mandarín, ¿verdad?

De manera similar, si quiero dar instrucciones a una computadora, necesito aprender un lenguaje que las computadoras puedan entender.

Ciertamente, **las computadoras no pueden entender las instrucciones dadas en ningún lenguaje humano.**

¿Tú qué piensas, qué idiomas entienden las computadoras?

*Anima al alumno a compartir cualquier lenguaje informático que pueda conocer o usar.*

**REA:** si/no.

**REA:** si.

**REA:** variada.

*El alumno puede mencionar JAVA, HTML, JavaScript, Python, etc.*

Vamos a aprender uno de esos lenguajes en el que podemos dar instrucciones a la computadora: **JavaScript**. De hecho, vamos a usar este lenguaje para dar instrucciones a la computadora para que diseñe un juego

para nosotros, cree aplicaciones móviles y ¡programe un satélite real también!

¡Sí, oíste bien!

PROGRAMAR UN SATÉLITE REAL en directo desde tu laptop al espacio exterior.

Antes de continuar, quiero preguntarte algo. Déjame compartir mi pantalla, con esto podrás ver lo que está pasando en mi pantalla.

### La maestra comienza a compartir pantalla



*Nota: comparte tu pantalla y abre [\[Actividad de la maestra 7 \(¿Sabías que...?\)\]](#) y haz las preguntas presentes en las diapositivas para generar entusiasmo.*





**IMPORTANTE**

*Nota: ¡haz las preguntas presentes en las diapositivas para generar entusiasmo y espera la respuesta!*

Sí, un “satélite”.

¿Y sabes qué? En nuestro **programa de exploración espacial**, podrás comunicarte y programar un satélite real, hacer clic en una imagen de la tierra, predecir el clima o los desastres naturales e incluso podrás rastrear la posición del satélite. Suena como un superpoder, ¿no es así?

**REA:** podemos ver NOTICIAS EN VIVO con la ayuda de un satélite.

## ¡¡PUNTO IMPORTANTE!!

**COMPRUEBA el tipo de anuncio que se muestra AL ALUMNO DE LA CLASE PRUEBA EN EL TABLERO.**

**Si el tipo de anuncio que se muestra = Roblox, verifica si el alumno ya ha instalado Roblox.**

**Si está instalado, realiza la actividad de Roblox a continuación, de lo contrario, continúa con el resto de las actividades de exhibición y programación.**



Actividad de Roblox

*Verifica el UTM del alumno.  
¿el niño se inscribió en una clase de prueba haciendo clic en un anuncio de Roblox?*



Actividad de Roblox

*En caso afirmativo, abre la actividad de la maestra 8 y juega Roblox.*

*Si no, omite la actividad de la maestra 8*

### ACTIVIDAD DIRIGIDA POR LA MAESTRA - 15 minutos

#### ACTIVIDAD

- Introducción a la plataforma [code.org](#).
- Comprender cómo escribir un programa para mostrar una figura rectangular.
- Aprender experimentando cómo cambiar las dimensiones y mover la figura.

#### Acción de la maestra

#### Acción del alumno

Vamos a comenzar aprendiendo a programar a través del diseño de juegos. Los juegos nos ofrecen muchos problemas interesantes y complejos para resolver como programadores.

Después de aprender a diseñar juegos, aprenderemos a crear apps móviles y luego aprenderemos a usar el aprendizaje automático (ML) y la inteligencia artificial (AI) en nuestros programas.

Comenzaremos con la construcción de un juego simple y luego construiremos progresivamente juegos más complejos, después de algunas clases.

Déjame mostrarte los primeros juegos que construiremos en el curso.

*La maestra muestra los enlaces de los 4 juegos.*

- [T-Rex Runner \[Actividad de la maestra 1\]](#)
- [Invasión pirata \[Actividad de la maestra 2\]](#)
- [Juego de carreras de autos multijugador \[Actividad de la maestra 3\]](#)

*¿Cómo jugar un juego de invasión pirata?*

- *Para jugar al juego de invasión pirata, usa las teclas de las flechas izquierda y derecha para cambiar el ángulo del cañón, y usa la flecha hacia abajo para disparar las balas de cañón a los barcos.*

*¿Cómo jugar un juego de carreras de autos multijugador?*

- *La primera vez que usas el juego de carreras de autos multijugador, debes ingresar tu nombre y generar una palabra secreta.*
- *Comparte la palabra secreta con el alumno.*
- *Tendrás que generar una nueva palabra secreta para cada clase.*

¡Antes de que podamos comenzar a programar, veamos la programación del juego de ruptura simple!

*El alumno observa.*

*La maestra puede abrir el [juego de romper ladrillos](#) [actividad de la maestra 4] para el alumno.*

**Nota 1:** a continuación, se muestra la imagen de la versión básica del juego de romper ladrillos.

**Nota 2:** haz clic en la pantalla para iniciar el juego con el movimiento de la pelota.



**Nota:** a continuación, se muestra la imagen de la versión completa del juego de romper ladrillos con más funciones, como puntajes y vidas, que los alumnos aprenderán a crear en las próximas clases. (Esta imagen solo se muestra como referencia para la maestra).



Incluso un juego simple como este tiene muchas cosas desafiantes para programar.

¿Puedes nombrar cuáles son los diferentes componentes de este juego simple?

*La maestra debe mostrar/hablar sobre las funcionalidades del juego que no se cubren en la respuesta del alumno.*

*Usa el siguiente texto para ayudarte a llenar los espacios en blanco.*

Si ves el juego, hay muchas cosas en él. Hay ladrillos y hay una paleta controlada por el mouse, hay una pelota que se mueve y la pelota rebota en las paredes, hay sonidos... ¡tantas cosas!

#### REA

- Hay ladrillos de colores en la parte superior que se mueven hacia abajo.
- Hay una pelota que rebota.
- Una vez que la pelota golpea los ladrillos en la parte superior, el ladrillo golpeado por la pelota desaparece.
- Hay puntos ganados cada vez que la

Una vez que sabemos lo que está pasando en el juego, ¿por dónde deberíamos empezar?

¡De repente, el juego parece ser complejo ahora!

Mientras programamos, a menudo tenemos que lidiar con problemas complejos. Abordamos problemas complejos dividiéndolos en varios problemas más pequeños y simples que podemos resolver.

Recuerda, cada vez que tengas que resolver un problema difícil y complejo, debes dividirlo en problemas más pequeños, cada uno de los cuales es lo suficientemente simple como para resolverlo solo.

¿Qué es lo más sencillo que podemos hacer en este juego?

Comencemos escribiendo el código para dibujar estos objetos.

Para escribir programas de computadora vamos a usar una plataforma en línea llamada [code.org](https://code.org).

*La maestra abre la [\[actividad de la maestra 5\]](#).*

pelota golpea el ladrillo.

- Hay una paleta del jugador que se mueve cada vez que movemos el mouse.

**REA:** variada.

**REA:** dibujar las formas de la paleta y la pelota.



*La maestra inicia sesión si ya tiene una cuenta en [code.org](#) o crea una cuenta de [code.org](#) cuando la use por primera vez.*

- Haz clic en "Iniciar sesión" en la esquina superior derecha.



- Haz clic en "**Crear una cuenta**", cuando uses [code.org](#) por primera vez y luego haz clic en **sesión**.



Usaremos la plataforma **laboratorio de juegos** de [code.org](#) para desarrollar juegos simples.

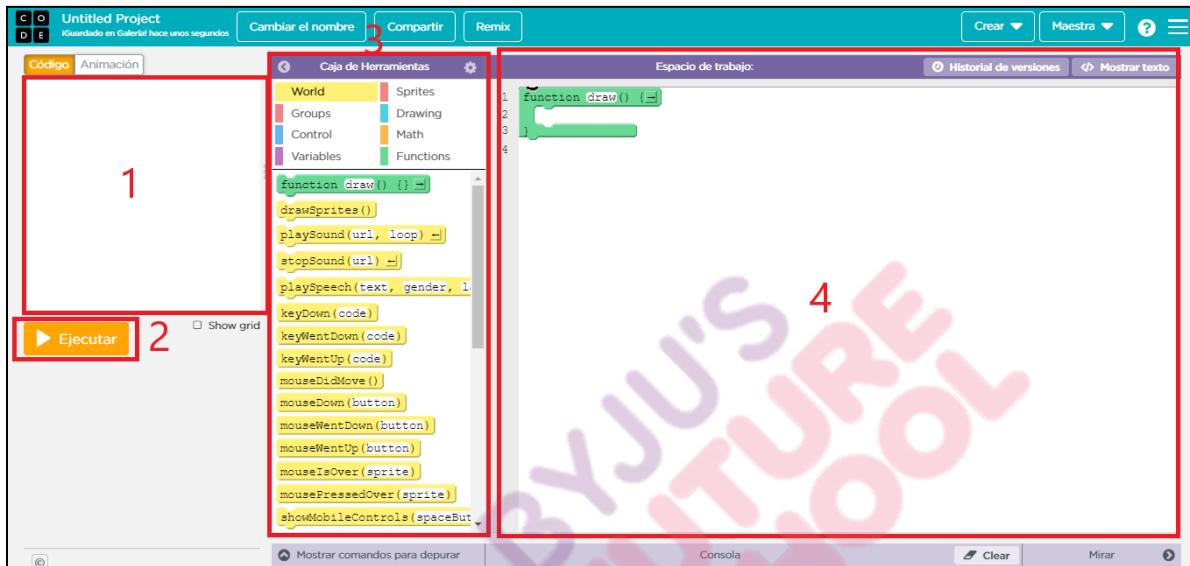
- En el tablero de [code.org](#):
  1. Haz clic en **Crear** (1 en la imagen a continuación).
  2. Haz clic en **laboratorio de juegos** (2 en la imagen a continuación).



The screenshot shows the 'Mi panel de control' (My control panel) of code.org. At the top, there's a 'Crear' (Create) button with a red box around it (labeled 1). On the right, there's a sidebar with various lab options: 'Laboratorio de sprites', 'Artista', 'Laboratorio de aplicaciones', 'Laboratorio de juegos' (which has a red box around it and is labeled 2), and 'Fiesta de baile'. Below the sidebar, there's a section for creating a new class ('Crea una sección') and another for 'Mis cursos' (My courses) showing 'Curso E (2021)'.

Entendamos la plataforma antes de que podamos escribir el código aquí.

**Nota:** La maestra muestra la interfaz de [code.org](#) (explica las secciones 1, 2, 3 y 4 en la imagen a continuación).



Comprensión de la interfaz de [code.org](#):

- Sección de la salida del código:** para ver el resultado del programa.
- Botón Ejecutar/Reiniciar:** el botón 'Ejecutar' se utiliza para iniciar el programa.

**Ejecutar**

Y el **botón 'Reiniciar'** se usa para detener el programa, si el programa ya se está ejecutando; esto se puede mostrar más tarde mientras se realiza la actividad.

**Reiniciar**

3. **Caja de herramientas:** la caja de herramientas es un cuadro de ayuda para usar algunos **bloques de código** directamente, que veremos más adelante. Podemos usar la flecha en la caja de herramientas para ocultarla/mostrarla.



4. **Espacio de trabajo:** se utiliza para escribir el código en formato de **texto o bloque**.

Estos son solo dos tipos diferentes de representación de las líneas de código en la plataforma [code.org](https://code.org).

#### Formato de texto:



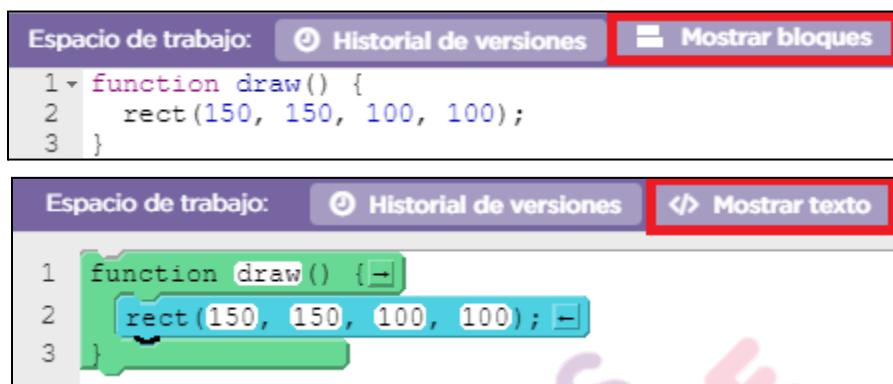
```
Workspace Version History Show Blocks
1 function draw() {
2   rect(150, 150, 100, 100);
3 }
```

#### Formato de bloque:



```
Workspace Version History Show Text
1 function draw() {
2   rect(150, 150, 100, 100);
3 }
```

**Nota:** Haz clic en "**Mostrar bloques**" si está en el modo de formato de texto para ver el formato de bloque del mismo código, o "**Mostrar texto**" si está en el modo de formato de bloque.



The image shows two screenshots of a Scratch workspace side-by-side. Both screenshots have a purple header bar with 'Espacio de trabajo:', a blue 'Historial de versiones' button, and a red 'Mostrar bloques' or 'Mostrar texto' button.

**Top Screenshot (Mostrar bloques mode):**

```

1 function draw() {
2     rect(150, 150, 100, 100);
3 }

```

**Bottom Screenshot (Mostrar texto mode):**

```

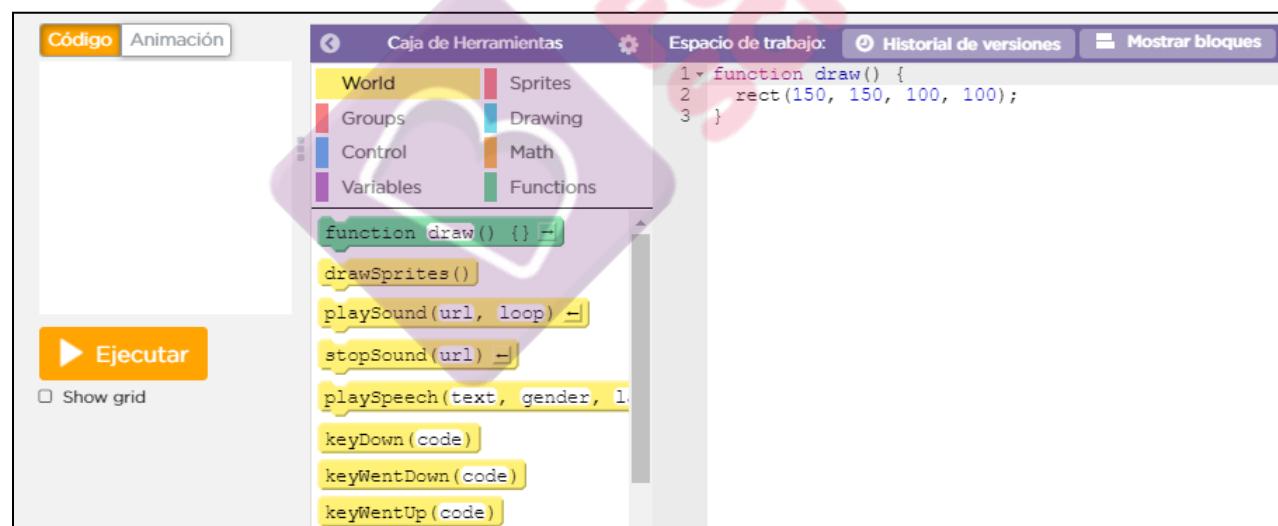
1 function draw() [ ]
2 rect(150, 150, 100, 100); [ ]
3 ]

```

Ahora vamos a empezar a escribir el código.

La maestra abre [la actividad de la maestra 6](#).

**Asegúrate de haber iniciado sesión para acceder al enlace del proyecto.**



The image shows a Scratch workspace with the following interface elements:

- Left Panel:** Shows tabs for 'Código' (selected) and 'Animación'. Below the tabs are buttons for 'Ejecutar' (Run) and 'Show grid'.
- Middle Panel:** A 'Caja de Herramientas' (Toolbox) with categories: World, Groups, Control, Variables, Sprites, Drawing, Math, and Functions. The 'World' category is highlighted.
- Right Panel:** The 'Espacio de trabajo' (Scratch Stage) with the code:

```

1 function draw() {
2     rect(150, 150, 100, 100);
3 }

```
- Header Bar:** Includes buttons for 'Historial de versiones' and 'Mostrar bloques'.

¿Recuerdas qué objetos de juego necesitamos para el juego de romper ladrillos?

¡Increíble!

**REA:** necesitamos **ladrillos**, una **pelota** y la **paleta**.

¿Puedes decirme cuál es la forma de esos objetos?

¡Magnífico!

Ahora, para dibujar un rectángulo necesitamos dar instrucciones a la computadora.

¿Qué crees que hace la instrucción **rect()**?

Sí. ¡Correcto!

¿Y ves un símbolo al final de la instrucción **rect()**? Esto es un punto y coma (;).

En JavaScript, un punto y coma (;) representa el final de una instrucción, como un punto al final de una oración en español.

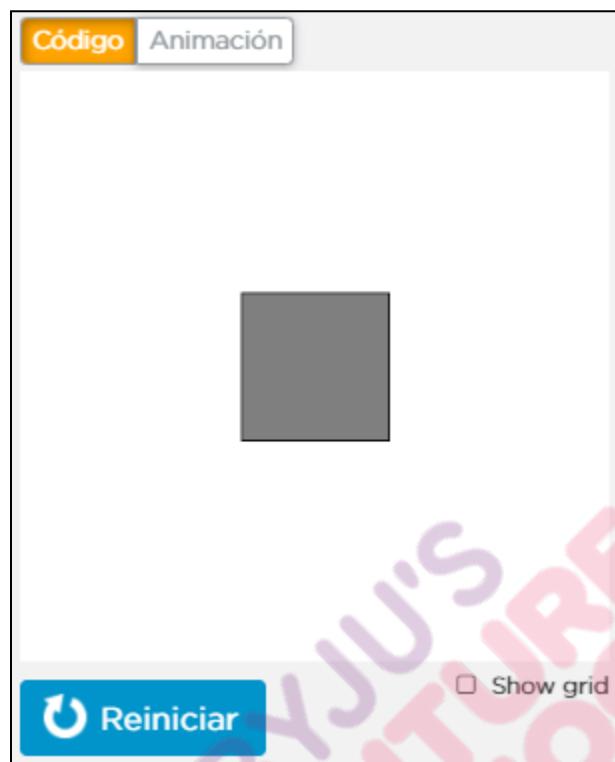
Aunque en JavaScript podremos ejecutar el programa sin punto y coma, en algunos lenguajes de programación las instrucciones no se ejecutarán sin el uso de punto y coma al final de las instrucciones.

Ahora vamos a ejecutar el código.

*Haz clic en el botón '**Ejecutar**' para ejecutar el código.*

**REA:** todos los objetos tienen forma de rectángulo.

**REA:** dibujará un rectángulo.



¡Estupendo!

Tenemos un rectángulo en la pantalla.

Pero queremos un rectángulo de diferentes tamaños para los ladrillos, la pelota y la paleta, ¿no?

¿Cómo hacemos eso?

¡Perfecto!

¿Puedes decirme qué significan estos números?

Ya sabes, lo mejor de la programación es que no tienes que esperar a que alguien más te dé las respuestas correctas.

**REA:** si.

**REA:** podemos cambiar los números dentro de la instrucción del rectángulo.

*El alumno adivina qué significan los números dentro de rect().*

Puedes experimentar, probar cosas, ver el resultado y encontrar las respuestas correctas por tu cuenta.

Experimentemos con estos números UNO POR UNO y tratemos de averiguar qué significan.

- Haz clic en el botón "Remix" para que el espacio de trabajo sea editable.

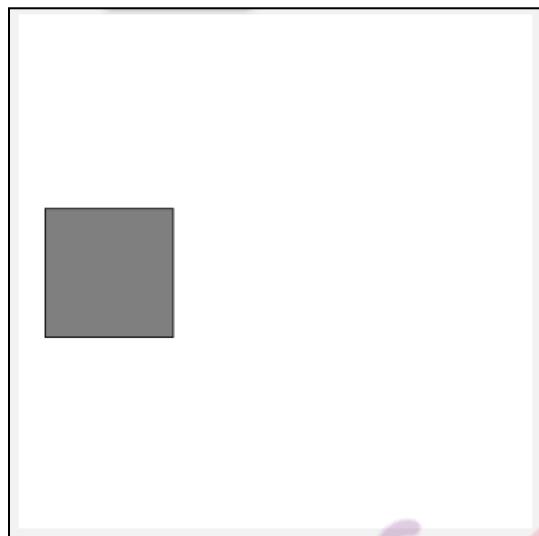


*El alumno da su opinión sobre cuál podría ser el primer número.  
Él reduce el primer número.*

*Toma la entrada que te diga el alumno, sobre cuál podría ser el primer número para cambiar el rectángulo.*

*La maestra ejecuta el código con el primer número reducido.*

```
function draw() {
    rect(20, 250, 10, 100);
}
```



¿Qué sucedió?

Aumentemos el número y veamos qué sucede.

¿Puedes darme un número más alto?

**REA:** el rectángulo se desplazó hacia la izquierda.

*El alumno da un número mayor para el primer número.*

```
function draw() {  
    rect(250, 250, 10, 100);  
}
```



¿Qué sucedió?

¿Qué crees que representa el primer número?

¡Sí! El primer número sirve para mover el rectángulo hacia la izquierda o hacia la derecha, o podemos decir en el eje x.

*La maestra repite los pasos anteriores para el segundo número.*

¿Qué crees que representa el segundo número?

*Cambia el segundo número de manera similar al primer número*

- *Reducelo primero y ejecuta el programa.*
- *Incrementalo y ejecuta el programa.*
- *Pídele al alumno que verifique si su suposición fue correcta.*

**REA:** el rectángulo se desplazó a la derecha.

**REA:** para colocar el rectángulo a la izquierda o a la derecha en la dirección horizontal.

**REA:** el segundo número se puede usar para mover el rectángulo hacia arriba y hacia abajo.

```
function draw() {  
    rect(250, 20, 10, 100);  
}
```

Código Animación



```
function draw() {  
    rect(250, 250, 10, 100);  
}
```

Código Animación



Experimentemos con los números tercero y cuarto, y veamos qué sucede con nuestro rectángulo.

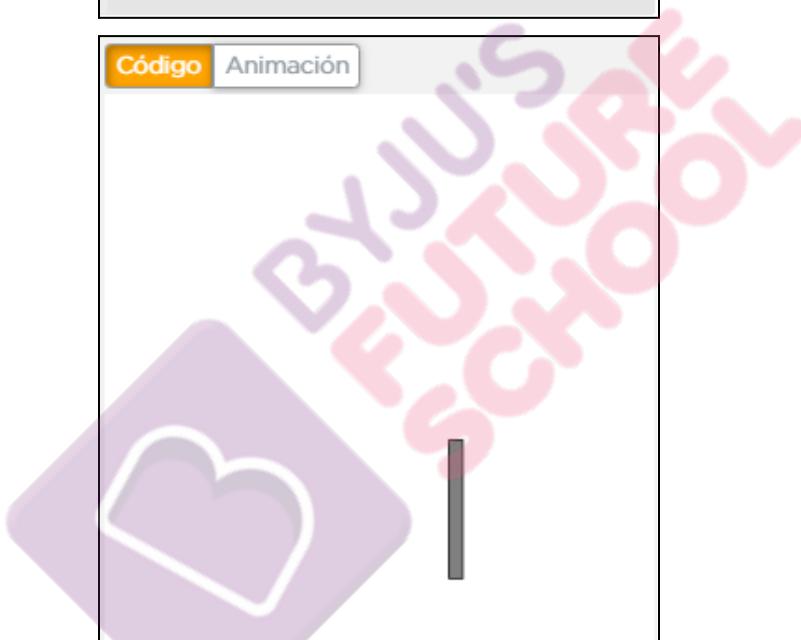
*La maestra repite los pasos anteriores con los números 3 y 4.*

**Asegúrate de revertir el tercer número antes de cambiar el cuarto número para mostrar la diferencia.**

*El alumno da información sobre qué números le gustaría probar para el tercer y cuarto número.*

```
function draw() {  
    rect(250, 250, 10, 100);  
}
```

Código Animación



```
function draw() {  
    rect(250, 250, 100, 10);  
}
```

Código Animación



¿Qué crees que representan los números 3 y 4?

¡Increíble!

Entonces, ahora sabemos cómo dibujar un rectángulo de diferentes tamaños y colocarlos en cualquier lugar de la pantalla. ¿No es así?

Comprender las posiciones x, y puede ser complicado a veces cuando colocamos objetos en la pantalla.

Comprendamos cómo se representan las posiciones x, y en una pantalla de computadora.

*La maestra hace clic en "Mostrar cuadrícula" debajo de la sección de la salida del código para mostrar los ejes x, y en el lienzo.*

**REA:** los números 3 y 4 representan el ancho y la altura del rectángulo.

**REA:** sí.



El (0,0) es el punto de inicio de las posiciones x, y. Está presente en la esquina superior izquierda de la sección de salida.

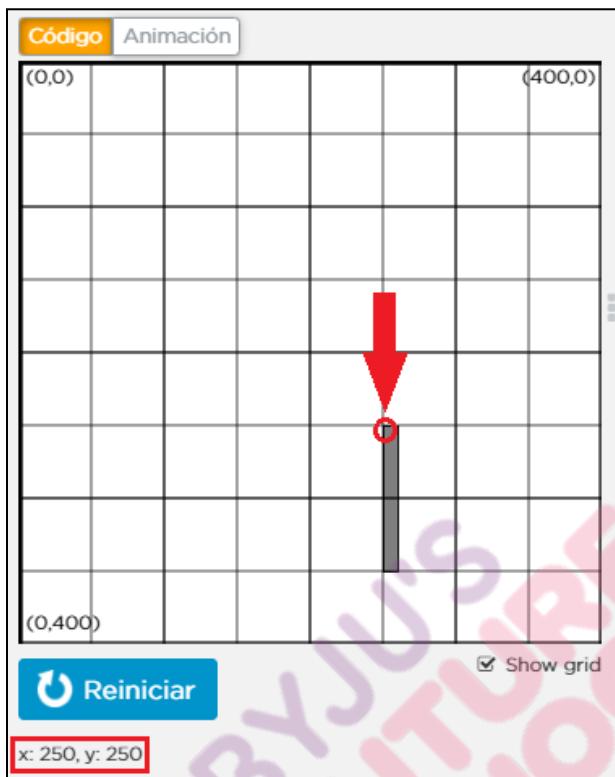
El valor de 'x' **aumenta de izquierda a derecha**.

El valor de 'y' **aumenta de arriba hacia abajo** (lo contrario de lo que aprendemos en matemáticas).

El (400, 400) es el punto final de las posiciones x, y. Éste se encuentra la esquina inferior derecha.

Esta sección de salida en code.org está limitada a un valor máximo de 400 para x, y.

Podemos mover el mouse sobre la sección de salida para averiguar los valores de x, y.



¿Crees que ahora puedes dibujar los 3 ladrillos, 1 pelota y la paleta solo?

¡Increíble!

Voy a dejar de compartir mi pantalla.

Comenzarás a compartir tu pantalla conmigo ahora, para que pueda ver lo que estás haciendo. Mientras dibujas los objetos, puedo ayudarte y guiarte.

¿Suena bien?

Solo asegúrate de pensar en voz alta para que pueda entender lo que está pasando en tu mente.

¡Empecemos!

*La maestra deja de compartir pantalla.*

**REA:** sí. Déjame intentarlo.

**REA:** sí.

*El alumno comienza a compartir pantalla.*

## La maestra deja de compartir pantalla

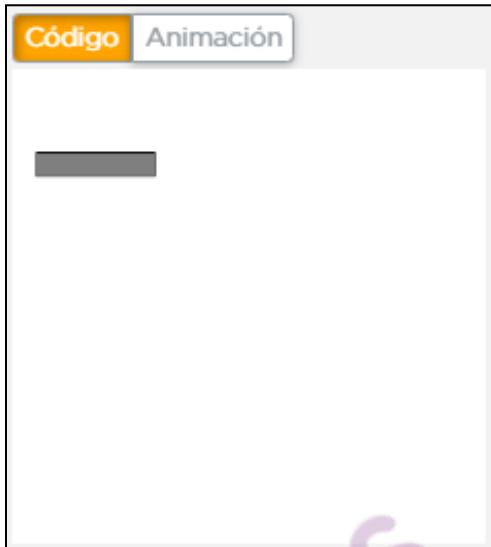
### ACTIVIDAD DIRIGIDA POR EL ALUMNO - 15 minutos

- Pide al alumno que presione ESC para regresar al panel.
- Guía al alumno para que comience a compartir pantalla.
- La maestra entra a pantalla completa.

### ACTIVIDAD

- Dibujar y colocar los ladrillos, la paleta del jugador y la pelota.
- Anima la paleta del jugador para que se mueva junto con el mouse.

Acción de la maestra	Acción del alumno
<p><i>El alumno puede pasar por los juegos dados en:</i></p> <p><a href="#"><u>Actividad del alumno 1 Juego del corredor T-Rex.</u></a></p> <p><a href="#"><u>Actividad del alumno 2 Invasión pirata.</u></a></p> <p><a href="#"><u>Actividad del alumno 3 Juego de carreras de autos multijugador.</u></a></p> <p><a href="#"><u>Actividad del alumno 4 Juego de romper ladrillos.</u></a></p>	
<p><i>Guía al alumno para que abra la <a href="#"><u>actividad del alumno 5.</u></a></i></p> <p><i>Pídele que inicien sesión o que cree una cuenta de code.org, usando su cuenta de Google.</i></p> <p>Un ladrillo se dibuja para ti en la parte superior.</p>	
<pre>function draw() {     rect(20, 70, 100, 20); }</pre>	



Ahora, tienes que:

- Dibujar los 2 ladrillos rectangulares más en la misma línea en la parte superior.
- Dibuja una paleta de jugador en la parte inferior y muévela con el mouse.
- Dibuja una pelota cuadrada en el centro.

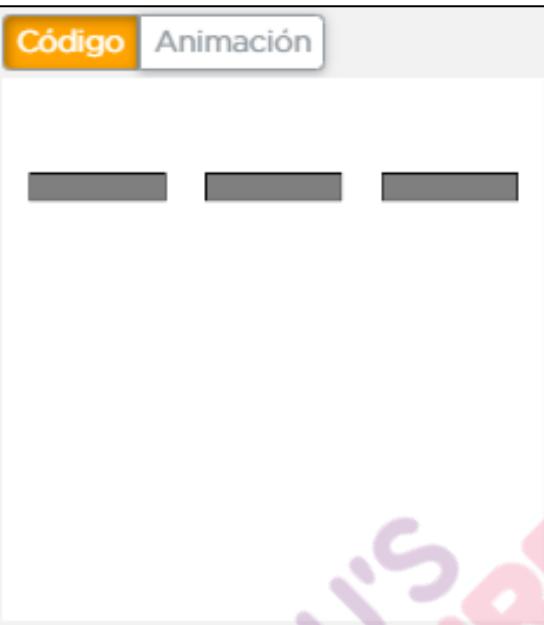
*Guía al alumno para que dibuje y coloque 2 ladrillos en la parte superior. Permite que el alumno experimente con los números.*

*Anima al alumno a escribir el código.*

*Déjalo ajustar el ancho y la altura de la paleta primero, para luego posicionarlos.*

**NO LES DES LOS NÚMEROS DE INMEDIATO.**

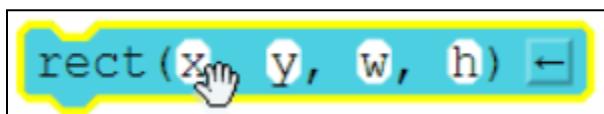
```
function draw() {  
    rect(20, 70, 100, 20);  
    rect(150, 70, 100, 20);  
    rect(280, 70, 100, 20);  
}
```



**Anima al alumno a escribir el código.**

**SI Y SOLO SI** el alumno tiene muchas dificultades para escribir el código en el modo de texto, guíalo para que siga los pasos a continuación para trabajar en el modo de bloque (toma la ayuda del [enlace](#) para comprender los pasos de trabajo):

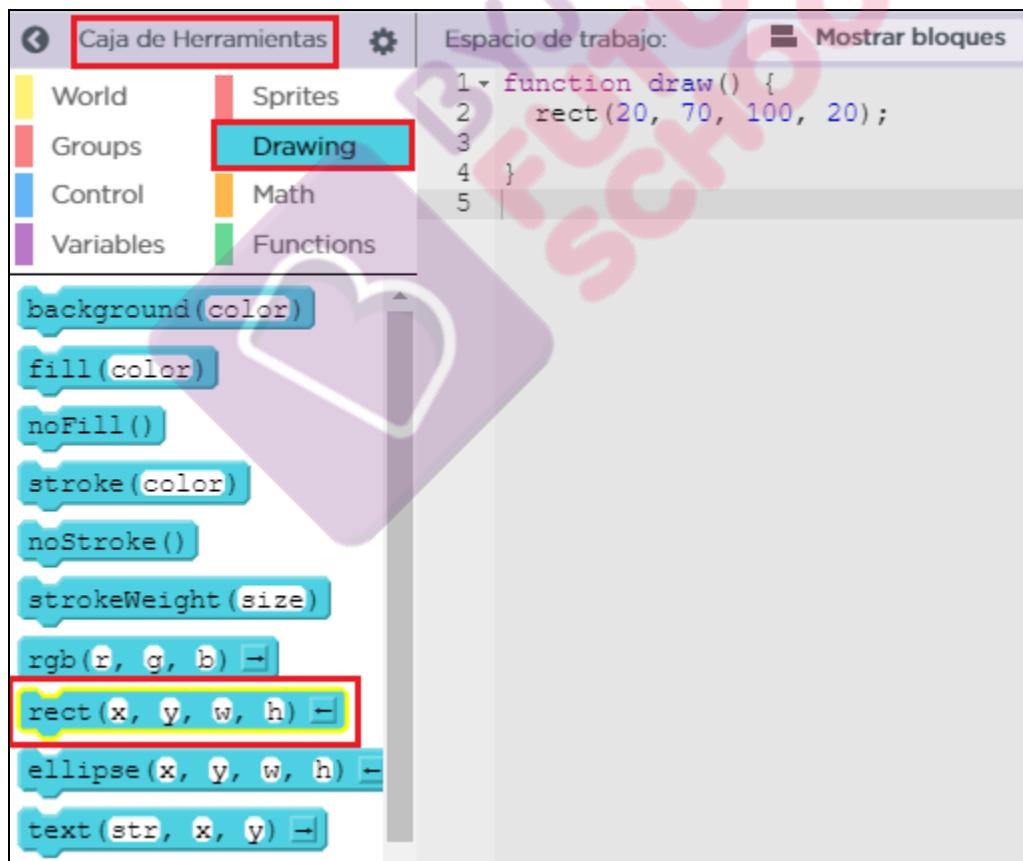
- Primero coloca el cursor en la línea apropiada dentro de la función draw().
- Haz clic en 'Drawing' - (dibujar), en la caja de herramientas.
- Desplázate dentro de la caja de herramientas para encontrar la instrucción **rect()**. Mientras te desplazas, verás el cursor como un símbolo de mano abierta.



- Haz clic y mantén presionado el mouse sobre **rect()**, dentro de la caja de herramientas, para seleccionar la instrucción. Mientras sostienes los bloques, el cursor se convierte en un puño (mano cerrada).

```
function draw() {
    rect(20, 70, 100, 20);
    rect(100, 100, 200, 200) ⏎
}
```

- Arrastra y suelta la instrucción **rect()** dentro de la función **draw()** sin dejar de presionar el clic del mouse antes de soltar.

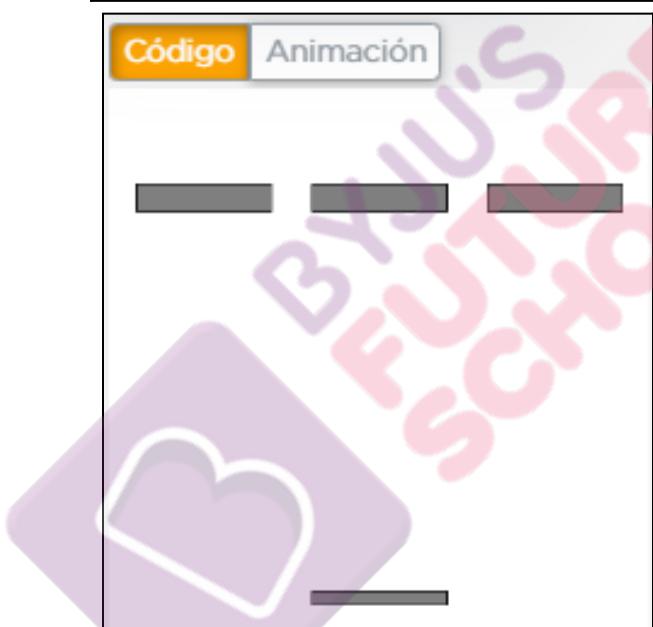


*Guíe al alumno para que dibuje y coloque la paleta izquierda del jugador controlada por computadora. Permite que el alumno experimente con los números.*

*NO LE DES LOS NÚMEROS DE INMEDIATO.*

```
function draw() {
    rect(20, 70, 100, 20);
    rect(150, 70, 100, 20);
    rect(280, 70, 100, 20);
    rect(150, 370, 100, 10);
}
```

Código      Animación



La paleta del jugador se dibuja en una posición fija en la dirección horizontal en este momento. ¿Cuál es su posición en esa dirección?

**REA:** es 150.

*Guíe al alumno para que responda la posición x de la paleta del jugador.*

En lugar de dibujar la paleta en una posición fija, queremos moverla horizontalmente (de izquierda a derecha y de derecha a izquierda) siempre que se mueva el mouse.

¿Puedes decirme qué posición de la paleta del jugador, x o y, debe cambiarse para moverla continuamente con el mouse en dirección horizontal?

Para dibujar al jugador en diferentes posiciones en la dirección horizontal, hay algo que almacena la posición horizontal (posición x) del mouse. Lo llamamos **mouseX**.

X representa al eje x (dirección horizontal).

En lugar de **150** para la posición x de la paleta del jugador, pongamos **mouseX** (o **World.mouseX**) allí para ver qué pasa.

**Nota:** *mouseX es una propiedad a la que se puede acceder a través de la sección World - (mundo) , que entenderemos en las próximas clases.*

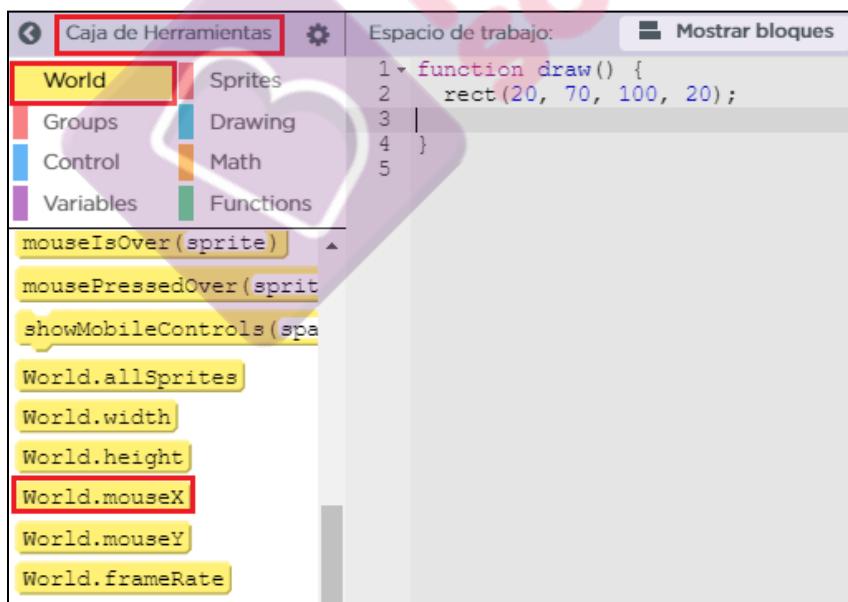
¿Qué crees que pasará si ejecutamos el código?

**REA:** deberíamos cambiar la posición x.

*El alumno reemplaza el 150 con mouseX (o World.mouseX).*

**REA:** la paleta se moverá con el mouse.

**Nota:** La instrucción **mouseX** se puede encontrar dentro de "World".

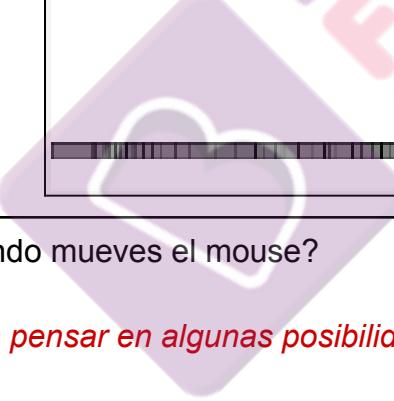


```
function draw() {
    rect(20, 70, 100, 20);
    rect(150, 70, 100, 20);
    rect(280, 70, 100, 20);
    rect(mouseX, 370, 100, 10);
}
```

```
function draw() {
    rect(20, 70, 100, 20);
    rect(150, 70, 100, 20);
    rect(280, 70, 100, 20);
    rect(World.mouseX, 370, 100, 10);
}
```

O

**Código** Animación



¿Qué sucede cuando mueves el mouse?

*Anima al alumno a pensar en algunas posibilidades.*

Bien. ¡Súper!

Pero esto no es lo que está pasando. ¡Sigue intentándolo!

Tratemos de resolverlo moviendo el mouse muy lentamente.

¿Qué sucede cuando mueves el mouse muy lentamente?

**REA:** cuando movemos el mouse, se dibujan múltiples líneas.

**REA:** un rectángulo se mueve y sigue dibujando varias líneas.

Pide al alumno que mueva la paleta muy lentamente y observe cómo se mueve el rectángulo cuando se mueve el mouse.

Mira más de cerca las múltiples líneas, ¿son líneas o múltiples rectángulos uno sobre el otro?

Sí. ¡Estupendo!

La paleta se mueve con el mouse, pero los viejos rectángulos todavía están en la pantalla. Cada vez que se mueve, se dibuja un nuevo rectángulo en la nueva posición del mouse sobre los rectángulos anteriores.

¿Por qué piensas que está sucediendo esto?

*Anima al alumno a pensar en algunas posibilidades.*

¿Puedes ver que hemos escrito cada instrucción dentro de la función **draw()**?

Cada línea de código dentro de la función **draw()** sigue ejecutándose repetidamente por nuestro programa. Eso significa que las 4 líneas en la función **draw()** se ejecutan una y otra vez.

Los 3 rectángulos en la parte superior son fijos, pero cada vez que movemos el mouse, se dibuja un nuevo rectángulo para la paleta del jugador.

Puedes imaginar la sección de salida como una hoja de dibujo, donde dibujaste algo. Ahora bien, si quieres volver a utilizar la misma hoja de dibujo, tendrás que borrar el dibujo antiguo o pintarlo con algún color para que quede limpio, ¿no?

¡Tendremos que hacer lo mismo también en nuestro programa!

**REA:** creo que estos son múltiples rectángulos.

**REA:** variada.

**REA:** si.

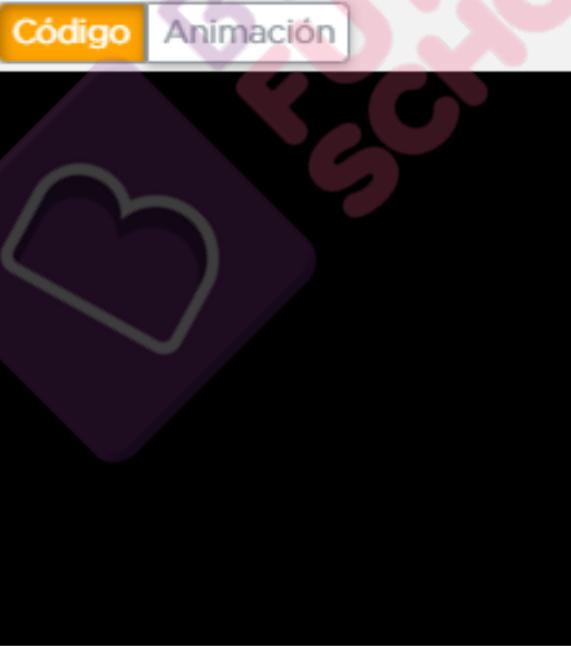
**REA:** si.

Entonces, ¿cómo eliminamos los rectángulos más antiguos?

Podemos pintar la pantalla de negro antes de dibujar cualquier cosa en la pantalla. De esta manera, pintaremos de negro sobre los rectángulos antiguos y luego dibujaremos uno nuevo en la parte superior.

*Guía al alumno para que use la instrucción **background('black')** - (fondo negro) debajo de las 4 instrucciones **rect()** y ejecute el programa.*

```
function draw() {  
    rect(20, 70, 100, 20);  
    rect(150, 70, 100, 20);  
    rect(280, 70, 100, 20);  
    rect(World.mouseX, 370, 100, 10);  
    background("black");  
}
```



**REA:** variada.

¿Qué sucedió?

¿Puedes tratar de pensar, por qué sucedió esto?

Veamos cómo escribimos el programa. Tenemos instrucciones en este orden:

1. **rect()**
2. **rect()**
3. **rect()**
4. **background()**

Aquí, el programa dibuja el primer rectángulo con la primera instrucción **rect()** y el segundo rectángulo con la segunda instrucción **rect()** y así sucesivamente. Después de eso, la instrucción **background()** hace el fondo negro al final. Y esto está sucediendo repetidamente. Por lo tanto, no podemos ver ningún rectángulo en ningún momento, ya que estos están pintados de negro cada vez que la computadora los dibuja.

¡Recuerda cómo discutimos al comienzo de la clase que el orden/secuencia de las instrucciones dadas a la computadora importa!

Entonces, ¿dónde crees que deberíamos escribir nuestra instrucción **background()**?

¡Magnífico!

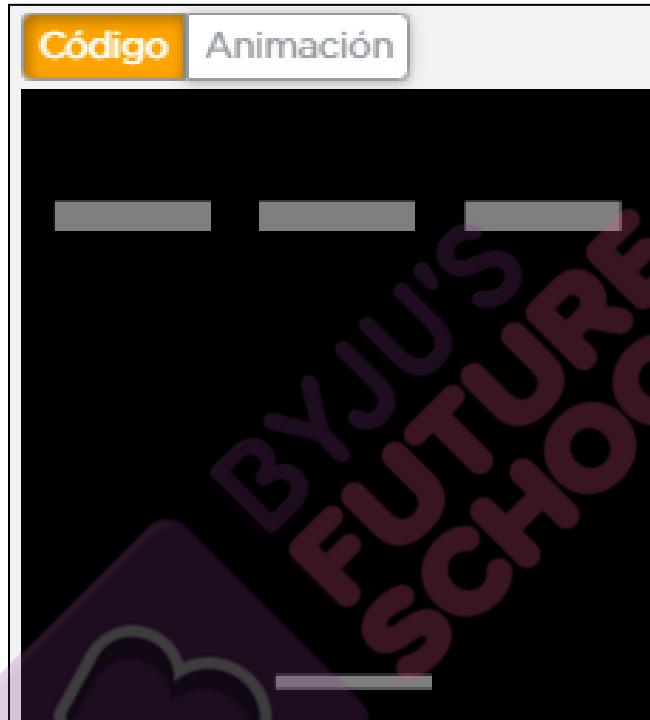
*Guía al alumno para que use la instrucción `background('black')` arriba de las 4 instrucciones `rect()` y ejecute el programa.*

**REA:** ¡toda la pantalla de salida se volvió negra!

**REA:** variada.

**REA:** deberíamos escribirlo antes de todas las instrucciones **rect()**.

```
function draw() {  
    background("black");  
    rect(20, 70, 100, 20);  
    rect(150, 70, 100, 20);  
    rect(280, 70, 100, 20);  
    rect(World.mousePosition, 370, 100, 10);  
}
```



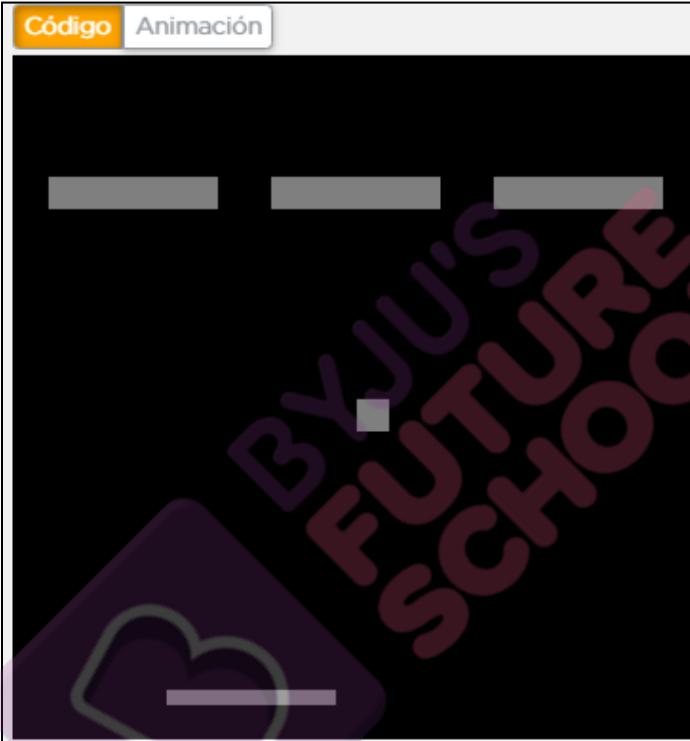
*Felicita al alumno.*

¡Súper! ¡Increíble!

La paleta del jugador ahora está animada. Se mueve junto con el ratón.

Ahora agreguemos rápidamente la pelota en el centro **(opcional, si queda tiempo extra)**.

```
function draw() {
    background("black");
    rect(20, 70, 100, 20);
    rect(150, 70, 100, 20);
    rect(280, 70, 100, 20);
    rect(World.mouseX, 370, 100, 10);
    rect(200, 200, 20, 20);
}
```



¡Hemos hecho bastante en una clase!

¡Impresionante <nombre del alumno>! ¡Eso es genial!

**Nota:** si el alumno aprende rápido, pídele que pruebe intente hacer las **ACTIVIDADES ADICIONALES**, a continuación en el documento.

En la esquina superior izquierda, presiona el ícono



Se abrirá una bandeja de emojis.



Presiona el emoticón de las gafas de sol.

## CONCLUSIÓN - 5 minutos

Repasemos rápidamente lo que hemos aprendido en la clase de hoy.

¿Puedes resumir rápidamente todo lo que has aprendido hoy?

#### REA:

- Es importante aprender a programar.
- Las computadoras entienden un lenguaje diferente, que puede dar instrucciones exactas.
- Aprendimos la instrucción `rect()` para dibujar un rectángulo.
- Aprendimos a mover figuras con el mouse.
- Dibujamos la paleta y la pelota.

¡Hiciste un excelente trabajo hoy!

Aprendimos a empezar a diseñar un juego hoy.

En la próxima clase, aprenderemos más sobre la programación del juego de romper ladrillos.

Mostrémosle esta app a tus padres.

Por favor, pregúntales si pueden unirse a nosotros un momento.

#### ***EL ALUMNO INVITA AL PADRE.***

#### Actividad dirigida por el alumno



Actividad de Roblox

¿Puedes hacer clic en [la actividad del alumno 8?](#)

¿Te gustaría crear juegos de este tipo?

#### [Actividad del alumno 8 - OBBY LLAMATIVO](#)

*Permite que el alumno juegue durante 5 minutos como máximo.*

	Crearemos juegos 3D y los publicaremos durante el curso. ¿Estás emocionado por esto?	<b>REA:</b> Sí  <b>REA:</b> Sí
--	--	--------------------------------------

### Llame a los padres, preséntate y celebra el logro del niño

**Maestra-** "Hola, me enorgullece decir que hoy, <nombre del niño> programó su primer juego. <Nombre del niño>, muestra el juego que creaste hoy".

- Aprecia y felicita al alumno por intentar y hacer un gran progreso para aprender a programar.
- Pregúntale qué le ha parecido la sesión.
- Repasa y comprueba su comprensión.
- Desafía al alumno a hacer que la pelota se mueva en el juego por su cuenta, después de la clase. No compartas ningún código, ni ayudes al alumno. Sigue alentando al alumno con afirmaciones como: '¿Qué crees que podemos hacer para mover la pelota, hacer que rebote, etc.?'; 'Parece una buena idea. Veamos qué sucede cuando haces esto'.

Permítanme darles un breve resumen de lo que aprenderemos en el curso.

Haz clic en el botón '**Iniciar presentación de diapositivas**' en el panel.



**Resumen del currículo**  
**Invita a los padres a ver el resumen del currículo.**

### Guía al alumno para que deje de compartir pantalla

#### RETROALIMENTACIÓN

- Aprecia al alumno por su esfuerzo durante la clase.
- Pide al alumno que haga notas para su diario de reflexiones, junto con el código que escribió en la clase de hoy.

<b>Acción de la maestra</b>	<b>Acción del alumno</b>
-----------------------------	--------------------------

Estamos comenzando en este emocionante viaje de programación. Espero ansiosamente enseñarte más en la próxima clase.

Mientras tanto, puedes experimentar con diferentes colores en la instrucción **background()** e intentar mover a la pelota.

Recuerda, en programación, podemos experimentar y aprender muchas cosas.

Para brindarles más detalles sobre el curso, así como responder cualquier pregunta que puedan tener, nuestro consejero académico se pondrá en contacto con usted.

**Si no pueden comunicarse con usted, ¿tiene un horario preferido?**

**<Anota la hora preferible en la que se podrá contactar a los padres para una mayor comunicación y actualiza esto con el consejero académico>**

Mi horario está casi lleno, pero me encantaría tener a su hijo como mi alumno, ¡ya que su hijo es excepcionalmente brillante con verdadero potencial de emprendimiento empresarial!

Gracias por su tiempo hoy. Por favor, quédense en el panel y no cierren esta página cuando finalice la clase. Nuestro plan de estudios completo junto con los detalles se mostrarán en el panel.

**Asegúrate de dar al menos 2 felicitaciones durante la clase por:**



La maestra hace clic en

**X Finalizar Clase**

## ACTIVIDADES ADICIONALES

**Actividad adicional 1**

Pregúntale al alumno qué pasa si el fondo se vuelve blanco.

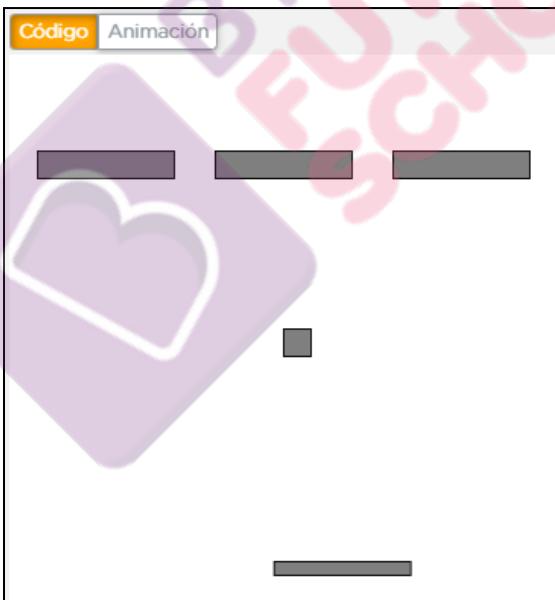
¿Crees que obtendremos múltiples rectángulos o un rectángulo moviéndose con el mouse?

*Espera la respuesta del alumno y luego deja que el alumno experimente.*

Obtendremos un rectángulo moviéndose con el mouse, ya que **no depende del color que elijamos dentro de la instrucción background()**.

**REA:** variada.

```
function draw() {  
    background("white");  
    rect(20, 70, 100, 20);  
    rect(150, 70, 100, 20);  
    rect(280, 70, 100, 20);  
    rect(World.mousePosition, 370, 100, 10);  
}
```



## Actividad adicional 2

**Guía al alumno para que pruebe la instrucción de relleno fill("ColorName")) - (llenar 'nombre del color').**

Podemos usar la instrucción **fill()** para llenar con colores dentro de los rectángulos.

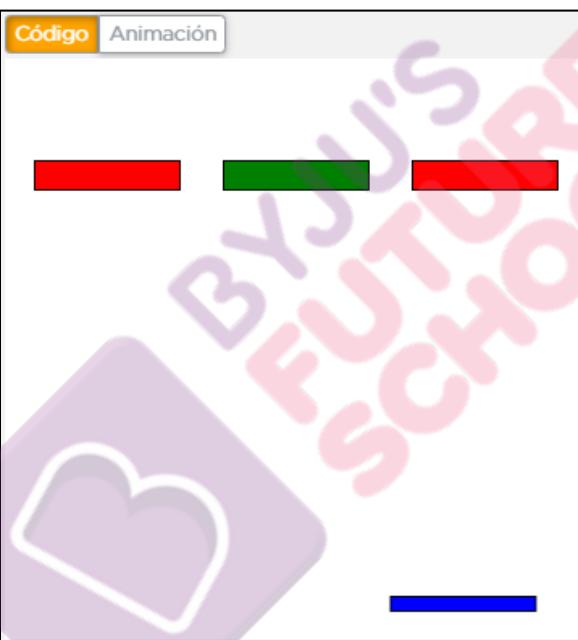
**Nota 1:** la instrucción **fill()** llena de colores todas las figuras dibujadas debajo de ella.

```
function draw() {  
    background("white");  
    rect(20, 70, 100, 20);  
    rect(150, 70, 100, 20);  
    fill("red");  
    rect(280, 70, 100, 20);  
    rect(World.mousePositionX, 370, 100, 10);  
}
```



**Nota 2:** utiliza la instrucción **fill()** encima de cada instrucción **rect()** si deseas llenar los rectángulos con los diferentes colores.

```
function draw() {  
  
    background("white");  
  
    fill("red");  
    rect(20, 70, 100, 20);  
  
    fill("green");  
    rect(150, 70, 100, 20);  
  
    fill("red");  
    rect(280, 70, 100, 20);  
  
    fill("blue");  
    rect(World.mouseX, 370, 100, 10);  
  
}
```



ENLACES DE LA ACTIVIDAD		
Nombre de la actividad	Descripción	Enlace
Actividad de la maestra 1	Juego T-Rex que corre	<a href="https://editor.p5js.org/BFS-MX-0537/full/ykd1waUVI">https://editor.p5js.org/BFS-MX-0537/full/ykd1waUVI</a>
Actividad de la maestra 2	Invasión pirata	<a href="https://byjusmx2.github.io/InvasionPirata/">https://byjusmx2.github.io/InvasionPirata/</a>
Actividad de la maestra 3	Juego de carreras de autos multijugador	<a href="https://github.com/karina-ramv/p5-multiplayer-car-race-game">https://github.com/karina-ramv/p5-multiplayer-car-race-game</a>
Actividad de la maestra 4	Juego de romper ladrillos	<a href="https://editor.p5js.org/BFS-MX-0537/full/s9Wd6gvz7">https://editor.p5js.org/BFS-MX-0537/full/s9Wd6gvz7</a>
Actividad de la maestra 5	Actividad de programación de la maestra	<a href="#">Actividad de programación de la maestra</a>
Actividad de la maestra 6	Referencia del código final	<a href="#">Referencia del código final</a>
Actividad de la maestra 7	¿Sabías qué...?	<a href="https://s3-whjr-curriculum-uploads.whjr.online/eefcff6f-2cd4-472c-a2f7-d999be316724.html">https://s3-whjr-curriculum-uploads.whjr.online/eefcff6f-2cd4-472c-a2f7-d999be316724.html</a>
Actividad de la maestra 8	Blazing-OBBY (juego de Roblox)	<a href="https://www.roblox.com/games/7351712185/Blazing-OBBY">https://www.roblox.com/games/7351712185/Blazing-OBBY</a>

Actividad de la maestra 9	Instrucciones para configurar Roblox en tu computadora	<a href="https://s3-whjr-curriculum-uploads.whjr.online/a07ef7cd-447a-4ac6-b323-c794bfa31dbd.pdf">https://s3-whjr-curriculum-uploads.whjr.online/a07ef7cd-447a-4ac6-b323-c794bfa31dbd.pdf</a>
Actividad de la maestra 10	Referencia del código final con actividad adicional	<a href="#">Referencia del código final con actividad adicional</a>
Actividad del alumno 1	Juego T-Rex que corre	<a href="https://editor.p5js.org/BFS-MX-0537/full/ykd1waUVI">https://editor.p5js.org/BFS-MX-0537/full/ykd1waUVI</a>
Actividad del alumno 2	Invasión pirata	<a href="https://byjusmx2.github.io/InvasionPirata/">https://byjusmx2.github.io/InvasionPirata/</a>
Actividad del alumno 3	Juego de carreras de autos multijugador	<a href="https://github.com/karina-ramv/p5-multiplayer-car-race-game">https://github.com/karina-ramv/p5-multiplayer-car-race-game</a>
Actividad del alumno 4	Juego de romper ladrillos	<a href="https://editor.p5js.org/BFS-MX-0537/full/s9Wd6gvz7">https://editor.p5js.org/BFS-MX-0537/full/s9Wd6gvz7</a>
Actividad del alumno 5	Actividad de programación del alumno	<a href="#">Actividad de programación del alumno</a>
Actividad del alumno 6	Blazing-OBBY	<a href="https://www.roblox.com/games/7351712185/Blazing-OBBY">https://www.roblox.com/games/7351712185/Blazing-OBBY</a>
Conclusión	Resumen del currículo	<a href="https://s3-whjr-curriculum-uploads.whjr.online/dc367d0f-b402-4252-96cf-ef857087cc1a.html">https://s3-whjr-curriculum-uploads.whjr.online/dc367d0f-b402-4252-96cf-ef857087cc1a.html</a>
Conclusión	Resumen del currículo (con notas para la maestra)	<a href="https://s3-whjr-curriculum-uploads.whjr.online/e8f01376-b12e-48db-aae3-51f006759cb9.html">https://s3-whjr-curriculum-uploads.whjr.online/e8f01376-b12e-48db-aae3-51f006759cb9.html</a>

## Glosario de referencia para maestras

No.	Abreviaturas / Término técnico	Explicación
1	AI (Inteligencia Artificial)	Se refiere a dar inteligencia humana a las máquinas que les permitan aprender y resolver problemas.
2	ML (Aprendizaje de máquina)	El aprendizaje de máquina es una aplicación de inteligencia artificial (IA) que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia sin estar programados explícitamente.
3	URL (Localizador de Recursos Uniforme)	Un localizador de recursos uniforme (URL). Una URL incorpora el nombre de dominio, junto con otra información detallada, para crear una dirección completa (o "dirección web") para dirigir un navegador a una página específica en línea llamada página web.
4	Nombre de dominio	El nombre de dominio es un componente de un localizador de recursos uniforme (URL) que se utiliza para acceder a sitios web, por ejemplo: URL: <a href="http://www.example.net/index.html">http://www.example.net/index.html</a> . Dominio de nivel superior: net. Dominio de segundo nivel: ejemplo.
5	Algoritmo	Un proceso o conjunto de reglas que se deben seguir en los cálculos u otras operaciones de resolución de problemas, especialmente por una computadora. También puedes considerar los algoritmos como un proceso paso a paso para resolver un problema.
6	React Native	Un lenguaje de programación basado en JS y HTML que se utiliza para crear aplicaciones móviles.
7	Boleta de calificaciones del alumno	Se utiliza para realizar un seguimiento del progreso de todos los alumnos. Que se separa como Competencia, Lógica, Creatividad y Concentración
8	Clases Capstone	Se trata de clases de repaso colocadas después de algunas clases como C8, que tienen la intención de repasar todos los conceptos que se han enseñado en clases anteriores.
9	Programa de Silicon Valley	Esta es una competencia entre los alumnos de Byju's FutureSchool, que tiene el desafío de crear una aplicación o

		juego que les permita resolver un problema de la vida real. Los mejores de ellos son seleccionados y recorridos por la NASA (virtualmente) y tienen la oportunidad de mostrar sus aplicaciones/juegos a varios emprendedores.
10	Canvas - ( <i>lienzo</i> )	Puede considerarse como el cuadro de visualización del output del editor.
11	Tecnología espacial	En estas clases utilizaremos datos del espacio para predecir eventos espaciales como la colisión de un asteroide con un planeta.
12	Motor físico	Un motor físico es un software de computadora que proporciona una simulación aproximada de ciertos sistemas físicos, como la dinámica de cuerpos rígidos (incluida la detección de colisiones), la dinámica de cuerpos flexibles y la dinámica de fluidos.
13	Juego invasión pirata	En este juego aprendemos el concepto de motor físico y lo usamos para crear un juego de invasión pirata, donde tenemos que defender nuestra torre derribando los barcos piratas.
14	Juego de carreras de autos multijugador	Aprendemos el concepto de base de datos en tiempo real usando este juego y entendemos cómo incluir la base de datos en nuestros juegos/aplicaciones y usamos esta lógica para crear un juego multijugador.
15	App estelar	Una app de tecnología espacial de React Native que ofrece actualizaciones sobre las constelaciones y los planetas visibles en el cielo, en cualquier noche, en una ubicación determinada.
16	App narración de historias	Usamos React Native para crear esta app. En esta app, el usuario puede agregar historias y la app leerá las historias convirtiendo texto a voz. El usuario también puede elegir entre un tema claro y oscuro.
17	App rastreador de la EEI	Esta app está hecha con react native, que rastrea la posición de la Estación Espacial Internacional y la muestra en el mapa. También da una advertencia visual a la tierra de los próximos meteoros.
18	App recomendación de películas	Esta app está construida con React Native, los patrones de visualización de películas del usuario de este estudio

		utilizando el aprendizaje automático y sugieren películas increíbles basadas en los datos del usuario.
19	App Exoplaneta	Un catálogo de nuestras estrellas vecinas celestiales, con datos e información sobre ellas.
20	Alimenta al conejo	En este juego usamos los conceptos del motor físico para conectar diferentes cuerpos. El objetivo del juego es romper la cuerda para que el conejito pueda comerse la fruta, mientras recolecta el máximo de estrellas posible.
21	Juego de romper ladrillos	Diseñamos esta app de juego en las primeras 7 clases, que nos ayudan a entender el movimiento simple del juego y la dinámica del mismo y también varias lógicas de programación.
22	REA	Respuesta esperada del alumno.



## Cómo triunfar en una clase de prueba



**Conectividad Y batería**

- ▶ Velocidad de internet de 20 Mbps
- ▶ Laptop siempre con batería
- ▶ respaldo de energía para enrutador



**Cámara y audífonos**

- ▶ Cámara encendida mientras das la clase
- ▶ Usa unos audífonos separados con micrófono incorporado
- ▶ Mantén el micrófono separado a 7 centímetros de tu boca



**Personalización de la clase**

- ▶ Llama a los estudiantes por sus nombres
- ▶ Da ejemplos relacionados a sus intereses y pasatiempos
- ▶ Mantenlos enganchados con la clase



**Comunicate mejor**

- ▶ Modera la velocidad de tu discurso
- ▶ Expresiones faciales y gestos con las manos
- ▶ Modula el tono para enfatizar cosas importantes



**Una mesa y una silla**

- ▶ Siéntate en una silla
- ▶ Mantén la laptop a 3.5 cm de ti
- ▶ Cámara de tu laptop por encima de los ojos



**Fondo y ambiente**

- ▶ Fondo limpio e de color claro
- ▶ Sala bien iluminada
- ▶ Fuente de luz suave a frente