





Tema	CONTROL DE EVENTOS	
Descripción de la clase	Ampliar los conceptos de control de eventos para integrar eventos keystroke a la aplicación web para llevar la experiencia de usuario al siguiente nivel.	
Clase	ADV-C86	
Duración	55 minutos	
Objetivo 	<ul style="list-style-type: none"> Desarrollar un juego web de Minecraft en 2D. 	
Recursos requeridos 	<ul style="list-style-type: none"> Recursos de la maestra <ul style="list-style-type: none"> Laptop o PC con webcam Cargador de laptop Internet de alta velocidad Credenciales de ingreso a gmail Audífonos con micrófono Bloque de notas y pluma Recursos del alumno <ul style="list-style-type: none"> Laptop o PC con webcam Credenciales de ingreso a gmail Audífonos con micrófono (opcional) Bloque de notas y pluma 	
Estructura de la clase	Rompiendo el hielo Actividad dirigida por la maestra Actividad dirigida por el alumno Conclusión Prácticas y sugerencias para el proyecto	5 minutos 10 minutos 30 minutos 5 minutos 5 minutos
ROMPIENDO EL HIELO - 5 mins		
Decir 		Hacer 

*Hola, me da gusto verte de nuevo. ¿Cómo has estado?
¿Estás emocionado de aprender algo nuevo?*

REA: Gracias, sí. Estoy muy emocionado.



La maestra inicia la presentación desde la diapositiva 1 a la 15
Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.



ACTIVIDAD DE LA MAESTRA - 10 mins

La maestra comienza a compartir pantalla

Decir



Hacer



NOTA: Abre la **Actividad de la maestra-5** y muéstrale la imagen al alumno. Dile que este es el resultado que esperan, pero que ellos pueden hacer lo que quieran utilizando su imaginación.

Abre la Actividad de la maestra-1 y muéstrale el sitio web al alumno.

***Nota:** Al principio, todas las teclas que presiones tardarán algunos segundos en responder, así que no es necesario que las presionen varias veces. Puede ser algo lento ya que debe cargar las imágenes, así que ten paciencia

Acciones por realizar:

1. Presiona las teclas de flecha del teclado una por una y muestra cómo se mueve el jugador.
2. Presiona la tecla **T** del teclado y muestra cómo aparece un bloque del **tronco**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Nosotros lo hemos utilizado para hacer el árbol.
3. Presiona la tecla **D** del teclado y muestra cómo viene un bloque **verde oscuro**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer el árbol.
4. Presiona la tecla **L** del teclado y muestra cómo aparece un bloque de color **verde claro**, el cual puede usarse para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer el árbol.
5. Presiona la tecla **G** en el teclado y muestra cómo aparece un bloque de **tierra**, el

cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer el suelo.

6. Presiona la tecla **W** en el teclado y muestra cómo aparece un bloque de **pared**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer la pared de la casa.
7. Presiona la tecla **Y** del teclado y muestra cómo aparece un bloque de **pared de color amarillo**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer la pared de la casa.
8. Presiona la tecla **U** en el teclado y muestra cómo aparece un bloque **único**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer un animal.
9. Presiona la tecla **R** en el teclado y muestra cómo aparece un bloque de **techo**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer el techo de una casa.
10. Presiona la tecla **C** en el teclado y muestra cómo aparece un bloque de **nube**, el cual puede ser utilizado para cualquier cosa según nuestra imaginación. Lo hemos utilizado para hacer una nube.
11. Presiona las teclas **shift** y **P** al mismo tiempo en el teclado, después presiona cualquiera de las teclas anteriores y muestra cómo el tamaño de los bloques se incrementa. También muestra que las etiquetas **Ancho** y **Altura** incrementan.

Ancho actual = 40 Altura actual = 40

12. Presiona las teclas **shift** y **M** al mismo tiempo en el teclado, después presiona cualquiera de las teclas anteriores y muestra cómo el tamaño de los bloques disminuye. También muestra que las etiquetas **Ancho** y **Altura** disminuyen.

Ancho actual = 30 Altura actual = 30

Nota: Abre la **Actividad de la maestra-4** y explícale lo siguiente al alumno.

1. El primer lienzo:

- No utiliza ninguna biblioteca y solo se utiliza javascript para obtener la imagen en el lienzo.

Nota: Mueve la imagen con las flechas como se menciona en la **Actividad de la maestra-4**. Muéstrale al alumno que cuando movemos al jugador, no desaparece la posición anterior. No queremos que esto suceda, ya que queremos un juego limpio cuando movemos al jugador.

- Debemos remover la posición anterior del jugador. Esto no se puede hacer con javascript normal, así que vamos a utilizar una biblioteca popular para construir juegos en un lienzo, la cual llamaremos **fabric.js**.
 - Es una biblioteca javascript. Tiene códigos javascript predefinidos y solo tenemos que llamarlos y utilizarlosPor ejemplo: Si queremos eliminar una imagen, solo debemos escribir **canvas.remove(NombreDeLaImagen)**. Esta función **remove()** está pre escrita en la biblioteca **fabric.js**.

2. El segundo lienzo:

- Esta utiliza la biblioteca **fabric.js**.

NOTA: Ahora vamos a mover al jugador al segundo lienzo con las teclas mencionadas en la **Actividad de la maestra-4**.

- Puedes ver la diferencia: Cuando mueves al jugador, la posición anterior se remueve y nos da una experiencia de juego limpia. Así es como utilizaremos la biblioteca **fabric.js** para construir el juego. Es muy sencillo utilizar **fabric.js**. Ya te hemos proporcionado el archivo, así que solo debes incluirlo como un archivo externo:

```
<script src="fabric.js"></script>
```

Este es el código HTML que se le proporciona al alumno:

```

<!DOCTYPE html>
<html>
<head>
  <title>C85</title>
  <link rel="stylesheet" href="style.css">
  <script src="fabric.js"></script>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
</head>
<body class="body_background">
  <center>

    <h2>MINECRAFT</h2>

    <div id="head_info">
      <h3>Ancho actual = </h3>
      <h3>Altura actual = </h3>
    </div>

  </center>
  <script src="main.js"></script>
</body>
</html>

```

El código HTML anterior contiene:

- Enlace para la hoja de estilo.
- Enlace para la biblioteca fabric.
- Enlace para las clases bootstrap.
- Etiqueta body.
- Etiqueta center.
- Algunas etiquetas de encabezado y etiquetas div.
- Enlace para el archivo main.js.

Comencemos con el código HTML.

Codigo HTML

1. Primero, debes **añadir una clase a la etiqueta body**.

- Añade la clase **body_background** a la etiqueta body, la cual añade un fondo a la página HTML.
- El código CSS ya está definido para esta clase en el archivo style.css. Solo debes añadir la clase en HTML:

```
<body class="body_background">
```

2. Ahora **añade la clase bootstrap para la etiqueta h2.**

→ **btn-primary** añadirá un color azul para el fondo y un color de letra blanco.

```
<h2 class="btn-primary">MINECRAFT</h2>
```

○ **Output:**

MINECRAFT

3. Ahora, **añade la clase bootstrap para la etiqueta div.**

→ **btn-danger** añadirá un color rojo para el fondo y un color de letra blanco.

```
<div id="head_info" class="btn-danger">
```

○ **Output:**

Ancho actual = Altura actual =

4. Ahora, **añade dos etiquetas span dentro de las etiquetas h3** y proporciona un id.

→ Estas etiquetas se utilizan para mostrar el ancho y alto de los bloques (como la pared, el tronco y otros) en la pantalla conforme incrementa o disminuye el tamaño.

→ Establece el texto a 30 para las dos etiquetas, ya que queremos que el tamaño inicial de los bloques sea de 30px.

```
<h3>Ancho actual = <span id="current_width">30</span></h3>  
<h3>Altura actual = <span id="current_height">30</span></h3>
```

○ **Output:**

Ancho actual = 30 Altura actual = 30

¡Muy bien! Terminamos con el código HTML.

Código JavaScript

Ahora vamos a escribir el código Javascript en el archivo **main.js**.

1. Primero debes **obtener la referencia del lienzo con fabric.Canvas()** y almacenarlo dentro de la variable **canvas**.

```
var canvas = new fabric.Canvas('myCanvas');
```

2. Ahora, **define el ancho y alto inicial de las imágenes de los bloques** y almacénalos en las variables.

- Vamos a establecer los valores para `block_image_width` y `block_image_height` como 30 y 30. Este número no es obligatorio, nosotros utilizamos estos para que se vean bien en el lienzo, pero tú puedes utilizar el valor que quieras

```
block_image_width = 30;  
block_image_height = 30;
```

3. Ahora, **define las coordenadas iniciales de "X", "Y" para el jugador** y almacénalas en las variables.

- Vamos a establecer los valores para `player_x` y `player_y` como 10 y 10. Este número no es obligatorio, nosotros utilizamos estos para que se comience en la parte superior del lienzo.

```
player_x = 10;  
player_y = 10;
```

4. Ahora define la variable `player_objetct` para **almacenar el objeto de la imagen del jugador**.

- En `fabric.js`, se almacenan las imágenes en el lienzo como objetos, así que conforme añadimos objetos, también podemos eliminarlos. Esto solo significa que podemos añadir y eliminar imágenes si las utilizamos como objetos.


```
var player_object= "";
```

5. Ahora añade la función `player_update()` para **añadir la imagen del jugador**. Esta función cargará la imagen del jugador en el lienzo.

```
function player_update()
{
    fabric.Image.fromURL("player.png", function(Img) {
        player_object = Img;

        player_object.scaleToWidth(150);
        player_object.scaleToHeight(140);
        player_object.set({
            top:player_y,
            left:player_x
        });
        canvas.add(player_object);
    });
}
```

Explicación del código:

- Define la función y establece el nombre `player_update()`.

```
function player_update()
{
```

- Define la función `fabric.Image.fromURL()` de `fabric.js` para subir una imagen al lienzo.

```
fabric.Image.fromURL("player.png", function(Img) {
```

- **fabric:** Este es el nombre de la biblioteca que utilizamos.

- **Image**: Esto indica que subimos una imagen.
- **fromURL**: Este contiene el enlace de la imagen y la función para subir la imagen.
- **“player.png”**: Esta es la imagen.
- **function(Img)**: Esta es la función que utilizamos para subir la imagen al lienzo.
 - **Img**: Este es el objeto de la imagen que se define por defecto.

El código anterior es un código preestablecido para añadir imágenes al lienzo.

→ Ahora almacena un objeto por defecto de la imagen dentro de nuestra variable.

- Cuando queramos eliminar, añadir, cambiar el ancho o alto de la imagen del jugador, nos vamos a referir a la variable **player_object**.

```
player_object = Img;
```

- Esta variable servirá de identificación para la imagen.
Por ejemplo: En HTML, tenemos una etiqueta img con el id:

Así que si queremos incrementar, disminuir o hacer cualquier modificación a la imagen en el CSS, debemos utilizar el id como se muestra a continuación:

```
#my_image  
{  
    width:100px;  
}
```

- Lo mismo ocurre con la variable **player_object** para manipular la imagen del jugador.

→ Añade el ancho de la imagen del jugador utilizando la variable **player_object**.

```
player_object.scaleToWidth(150);
```

- **player_object** es el objeto de la imagen.
- **scanToWidth(150)** establece el ancho de la imagen.
 - **150** es el ancho que queremos establecer. Este número no es obligatorio. Puede ser cualquier número, pero nosotros utilizamos 150 para que se vea bien en el lienzo..

→ Añade el alto de la imagen del jugador utilizando la variable **player_object**..

```
player_object.scaleToHeight(140);
```

- **player_object** es el objeto de la imagen.
- **scanToHeight(140)** establece el alto de la imagen.
 - **140** es el alto que queremos establecer. Este número no es obligatorio. Puede ser cualquier número, pero nosotros utilizamos 150 para que se vea bien en el lienzo.

→ Establece las coordenadas "X", "Y" para la imagen.

```
player_object.set({  
  top:player_y,  
  left:player_x  
});
```

- **player_object** es el objeto de la imagen
- **set** establece las coordenadas "X", "Y" de la imagen del jugador.
- **top** es la coordenada "Y":
// Vamos a establecer la coordenada "Y" para el valor player_y, el cual tiene el número 10 que definimos antes.
- **left** es la coordenada "X".
// Vamos a establecer la coordenada "X" para el valor player_x, el cual tiene el número 10 que definimos antes.

→ Añade la imagen al lienzo.

```
canvas.add(player_object);
```

- **canvas** es la variable que creamos al inicio que contiene el elemento canvas.
- **add(player_object)**: Esta es la función para añadir algo al lienzo.
 - **player_object** es el objeto de la imagen del jugador.

Como pudiste ver en la [Actividad de la maestra-1](#), cuando presionamos cualquier tecla, algunos objetos aparecen en el lienzo, como paredes, paredes amarillas, bloques verdes, entre otros.

Para poder añadir todas estas imágenes, debemos escribir una función para añadir imágenes específicas según la tecla que se oprima.

6. Añade la función add para **añadir diferentes imágenes según la tecla oprimida**.

```
function new_image(get_image)
{
    fabric.Image.fromURL(get_image, function(img) {
        block_image_object = img;

        block_image_object.scaleToWidth(block_image_width);
        block_image_object.scaleToHeight(block_image_height);
        block_image_object.set({
            top:player_y,
            left:player_x
        });
        canvas.add(block_image_object);

    });
}
```

Explicación del código:

→ Define la función new_image().

```
function new_image(get_image)
{
```

- Dentro de los paréntesis, vamos a añadir la variable **get_image**.
- Esta variable contiene la imagen del bloque correspondiente a la tecla que se oprime.

Por ejemplo:

Si se presiona la tecla **Y**, la variable tendrá la imagen de una pared amarilla.

Si se presiona la tecla **D**, la variable tendrá la imagen de un bloque verde oscuro..

→ Ahora define la función de fabric.js `fromURL()` para subir las imágenes al lienzo.

```
fabric.Image.fromURL(get_image, function(Img) {
```

- **fabric**: Este es el nombre de la biblioteca que utilizamos.
- **Image**: Esto indica que subimos una imagen.
- **fromURL**: Este contiene el enlace de la imagen y la función para subir la imagen.
- **get_image**: Esta es la variable que contiene la imagen correspondiente a la tecla que se oprime.
- **function(Img)**: Esta es la función que utilizamos para subir la imagen al lienzo.
 - **Img**: Este es el objeto de la imagen que se define por defecto.

→ Ahora almacena un objeto por defecto de la imagen dentro de nuestra variable. Cuando queramos cambiar el ancho o alto de la imagen, nos vamos a referir a la variable **block_image_object**.

```
block_image_object = Img;
```

- Esta variable servirá de identificación para la imagen.
Por ejemplo: En HTML, tenemos una etiqueta `img` con el id:

Así que si queremos incrementar, disminuir o hacer cualquier modificación a la imagen en el CSS, debemos utilizar el id como se muestra a continuación:
#my_image
{
width:100px;

```
}
```

- Lo mismo ocurre con la variable **block_image_object** para manipular la imagen del bloque.

→ Añade el ancho de la imagen del bloque utilizando la variable **block_image_object**.

```
block_image_object.scaleToWidth(block_image_width);
```

- **block_image_object** es el objeto de la imagen.
- **scaleToWidth(block_image_width)** establece el ancho de la imagen.
- **block_image_width**: Vamos a establecer el ancho del valor **block_image_width**, el cual es 30 como lo definimos anteriormente.

→ Añade el alto de la imagen del bloque utilizando la variable **block_image_object**.

```
block_image_object.scaleToHeight(block_image_height);
```

- **block_image_object** es el objeto de la imagen.
- **scaleToHeight(block_image_height)** establece el ancho de la imagen.
- **block_image_height**: Vamos a establecer el ancho del valor **block_image_height**, el cual es 30 como lo definimos anteriormente.

→ Establece las coordenadas "X", "Y" para la imagen.

```
block_image_object.set({  
  top:player_y,  
  left:player_x  
});
```

- **block_image_object** es el objeto de la imagen
- **set** establece las coordenadas "X", "Y" de la imagen del bloque.

- **top** es la coordenada "Y".
// Vamos a establecer la coordenada "Y" para el valor player_y, el cual tiene el número 10 que definimos antes.
*Vamos a establecer el valor de player_y, ya que queremos cargar una imagen donde se mueve el jugador.
- **left** es la coordenada "X".
// Vamos a establecer la coordenada "X" para el valor player_x, el cual tiene el número 10 que definimos antes.
*Vamos a establecer el valor de player_x, ya que queremos cargar una imagen donde se mueve el jugador.

→ Añade la imagen al lienzo con la función canvas.add().

```
canvas.add(block_image_object);
```

- **canvas** es la variable que creamos al inicio que contiene el elemento canvas.
- **add(block_image_height)**: Esta es la función para añadir algo al lienzo.
- **block_image_height** es el objeto de la imagen del jugador.

La maestra deja de compartir pantalla

ACTIVIDAD DEL ALUMNO  - 30 mins

Ahora es tu turno.

- **Pídele al alumno presionar la tecla ESC para regresar al panel**
- **Pídele al alumno Compartir Pantalla**
- **La maestra debe usar el modo de Pantalla Completa**

El alumno comienza a compartir pantalla

***Nota: Sigue los siguientes pasos.**

***Nota: Deja que el alumno copie y pegue el código que se repite.**

Pídele al alumno que descargue la carpeta **minecraft** de la **Actividad del alumno-2**. Esta contiene parte del código

Actividad del alumno-1 contiene el diagrama de código para el alumno.

Actividad del alumno-2 contiene el código.

HTML, los archivos style.css, fabric.js y un archivo main.js vacío.

El alumno debe añadir los siguientes elementos HTML al archivo:

- Añadir una clase **body_background** a la etiqueta body.
- Añadir **btn-primary** a la etiqueta h2.
- Añadir **btn-danger** al div con id="head_info".
- Añadir **etiquetas span tag** para el ancho y alto.

Comienza a escribir el código javascript en el archivo **main.js**.

Deja que el alumno escriba el código como se explicó anteriormente.

Actividad del alumno-3
contiene un ejemplo de **fabric.js**

Anima al alumno y ayúdalo en caso de que sea necesario.

***Nota: El alumno debe descargar la carpeta [minecraft](#) de la Actividad del alumno-2. Esta contiene parte del código HTML, los archivos style.css, fabric.js y un archivo main.js vacío. El alumno debe añadir el código HTML en el archivo HTML y comenzar a programar el código JavaScript en el archivo main.js.**

1. Añade la clase en la etiqueta body:

```
<body class="body_background">
```

2. Añade la clase bootstrap en la etiqueta h2:

```
<h2 class="btn-primary">MINECRAFT</h2>
```

3. Añade la clase bootstrap en la etiqueta h3:

```
<h3>Ancho actual = <span id="current_width">30</span></h3>
<h3>Altura actual = <span id="current_height">30</span></h3>
```

Código en el archivo main.js

4. Crea el elemento canvas:

```
var canvas = new fabric.Canvas('myCanvas');
```

5. Define el ancho y el alto de la imagen del bloque.


```
block_image_width = 30;  
block_image_height = 30;
```

6. Define las coordenadas "X", "Y" de la imagen del jugador.

```
player_x = 10;  
player_y = 10;
```

7. Añade la función player_update().

```
function player_update()  
{  
    fabric.Image.fromURL("player.png", function(Img) {  
        player_object = Img;  
  
        player_object.scaleToWidth(150);  
        player_object.scaleToHeight(140);  
        player_object.set({  
            top:player_y,  
            left:player_x  
        });  
        canvas.add(player_object);  
    });  
}
```

8. Añade la función para añadir una imagen de bloque.

```
function new_image(get_image)
{
    fabric.Image.fromURL(get_image, function(Img) {
        block_image_object = Img;

        block_image_object.scaleToWidth(block_image_width);
        block_image_object.scaleToHeight(block_image_height);
        block_image_object.set({
            top:player_y,
            left:player_x
        });
        canvas.add(block_image_object);

    });
}
```

Guarda bien los archivos con los códigos ya que deberán subirse a github una vez que terminen de crear el sitio web.

La maestra le pide al alumno dejar de Compartir Pantalla

CONCLUSIÓN - 5 mins



La maestra inicia la presentación desde la diapositiva 16 a la 18
Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Lo hiciste muy bien. Felicidades por tu excelente trabajo.

Felicita al alumno por su trabajo.

(Da al menos dos felicitaciones)

Presiona el icono de felicitaciones para **Resolver**

	<p>Creativamente las Actividades</p>  <p>Presiona el icono de felicitaciones para Muy Buena Pregunta</p>  <p>Presiona el icono de felicitaciones para "!Te concentraste!"</p> 
<p>La maestra comienza a compartir pantalla</p>	
<p>SIGUIENTE CLASE Y SUGERENCIAS PARA EL PROYECTO - 5 mins</p>	
<p> La maestra inicia la presentación desde la diapositiva 19 a la 21 Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.</p>	
<p>Decir</p> 	<p>Hacer</p> 
<p>FELICITACIÓN DE CUMPLEAÑOS</p> <p>Objetivo del proyecto:</p> <p>En la clase 86, aprendiste a utilizar la biblioteca fabric.js y las funciones dentro de ella que utilizaste para añadir imágenes al lienzo.</p> <p>Para este proyecto, deberás concentrarte en subir imágenes al lienzo con la ayuda de fabric.js.</p>	<p>Abre el enlace con la solución del proyecto para mostrársela al alumno.</p>

Historia

Felicita a tus seres queridos y sorpréndelos diseñando y creando una tarjeta de cumpleaños. Hecha por ti con tus habilidades de programación. Diseña una página, crea un elemento de lienzo utilizando fabric.js y, a después, carga la imagen de la tarjeta de cumpleaños en el lienzo y reproduce la música cuando se haga clic en el botón.

Estoy muy emocionada por ver cómo haces este proyecto. Sé que lo harás muy bien.

¡Buena suerte!

La maestra deja de compartir pantalla

La maestra hace clic en

✕ Finalizar Clase

Actividad	Nombre de la actividad	Enlaces
Actividad de la maestra 1	ENLACE DEL SITIO WEB	https://amdavalos.github.io/minecraft/
Actividad de la maestra 2	DIAGRAMA DEL CÓDIGO	https://s3-whjr-curriculum-uploads.whj.r.online/a64a707b-16c2-4b0e-8495-07d04266bed8.pdf
Actividad de la maestra 3	FUENTE DEL CÓDIGO	https://drive.google.com/drive/folders/1xkmJffjXj9w-FVULnf7OriywpMo2Gvg?usp=sharing
Actividad de la maestra 4	EJEMPLO DE FABRIC.JS	https://amdavalos.github.io/demo-fabric/
Actividad de la maestra 5	IMAGEN DEL RESULTADO	https://drive.google.com/file/d/16yFFto5J8LvM4nri3PCD2mz4GS-Vti9w/view

		?usp=sharing
Actividad del alumno 1	DIAGRAMA DEL CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/a64a707b-16c2-4b0e-8495-07d04266bed8.pdf
Actividad del alumno 2	FUENTE DEL CÓDIGO	https://drive.google.com/drive/folders/1xkmJffjXj9w-FVULnf7OriywpMo2GvgY?usp=sharing
Actividad del alumno 3	EJEMPLO DE FABRIC.JS	https://amdavalos.github.io/demo-fabric/
Solución del proyecto	DESEO DE CUMPLEAÑOS	https://amdavalos.github.io/C86Proyecto/
Referencia de la maestra: Apoyo visual	Apoyo visual con notas	https://s3-whjr-curriculum-uploads.whjr.online/90e867d5-64d1-4149-a932-01cb59285218.html
Referencia de la maestra: Quiz de la clase	Quiz de la clase	https://s3-whjr-curriculum-uploads.whjr.online/560e9dbc-0a05-408e-b503-5297fbd4443.pdf

Referencia de la maestra-NO DEBE AÑADIRSE PANEL DE ENLACE ACTIVIDAD		
Referencia de la maestra	Apoyo visual sin notas	https://s3-whjr-curriculum-uploads.whjr.online/d8bbd05d-0814-4d69-8178-ae4a22de1473.html