


Tema	IDENTIFICACIÓN DEL DIBUJO	
Descripción de clase	Los alumnos se aventuran en el mar de las redes neuronales pre-entrenadas y las estudian para conocer sus posibles aplicaciones de interés. Aprender los métodos de la red neuronal fortalece su confianza para examinar el código complejo y aplicar solo lo necesario para resolver un problema. Manejar la complejidad es esencial en la IA (Inteligencia Artificial).	
Clase	ADV-C118	
Duración de la clase	55 minutos	
Objetivo 	<ul style="list-style-type: none"> • Agregar estilo a los elementos HTML. • Iniciar la programación Javascript. 	
Recursos requeridos	<ul style="list-style-type: none"> • Recursos para maestras: <ul style="list-style-type: none"> ○ Computadora portátil / PC con cámara web. ○ Utiliza las credenciales de inicio de sesión de Gmail. ○ Auricular con micrófono. ○ Bloc de notas y bolígrafo. • Recursos para alumnos: <ul style="list-style-type: none"> ○ Computadora portátil / PC con cámara web. ○ Utiliza las credenciales de inicio de sesión de Gmail. ○ Auricular con micrófono (opcional). ○ Bloc de notas y bolígrafo. 	
Estructura de la clase	Rompiendo el hielo Actividad dirigida por la maestra Actividad dirigida por alumnos Conclusión Pistas y sugerencias para el proyecto	5 minutos 10 minutos 30 minutos 5 minutos 5 minutos

ROMPIENDO EL HIELO - 5 minutos



La maestra inicia la presentación desde la diapositiva 1 a la 12.
Consulta las notas en tu pantalla y sigues las instrucciones de cada diapositiva



ACTIVIDAD DE LA MAESTRA - 10 minutos

La maestra comienza a compartir pantalla

Decir



Hacer



Recuerda, no tienes que explicar todo el código JS. Simplemente explica los conceptos según el documento.

Sigue el flujo de la clase:

1. Primero, explica el código CSS y JS.
2. Luego, deja que el alumno haga el código HTML, CSS y JS.
3. Luego, si tienes tiempo, realiza las actividades adicionales.

En la clase anterior, hicimos el código HTML en el archivo [index.html](#):

```
<body>
  <center>
    <!-- Modal -->
    <div id="myModal" class="modal fade">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&times;</button>
          <h4 class="modal-title">Dibujos</h4>
        </div>

        <div class="modal-body">
          
          
        </div>
      </div>
    </div>

    <h1>Identificación de dibujos
    <button id="show_sketch" class="btn btn-info" data-toggle="modal" data-target="#myModal">Mostrar dibujos</button>
    </h1>
    <p id="label">Etiqueta: </p>
    <p id="confidence">Confianza: </p>
    <button class="btn btn-danger btn-lg" onclick="clearCanvas()">Borrar lienzo</button>
  </center>

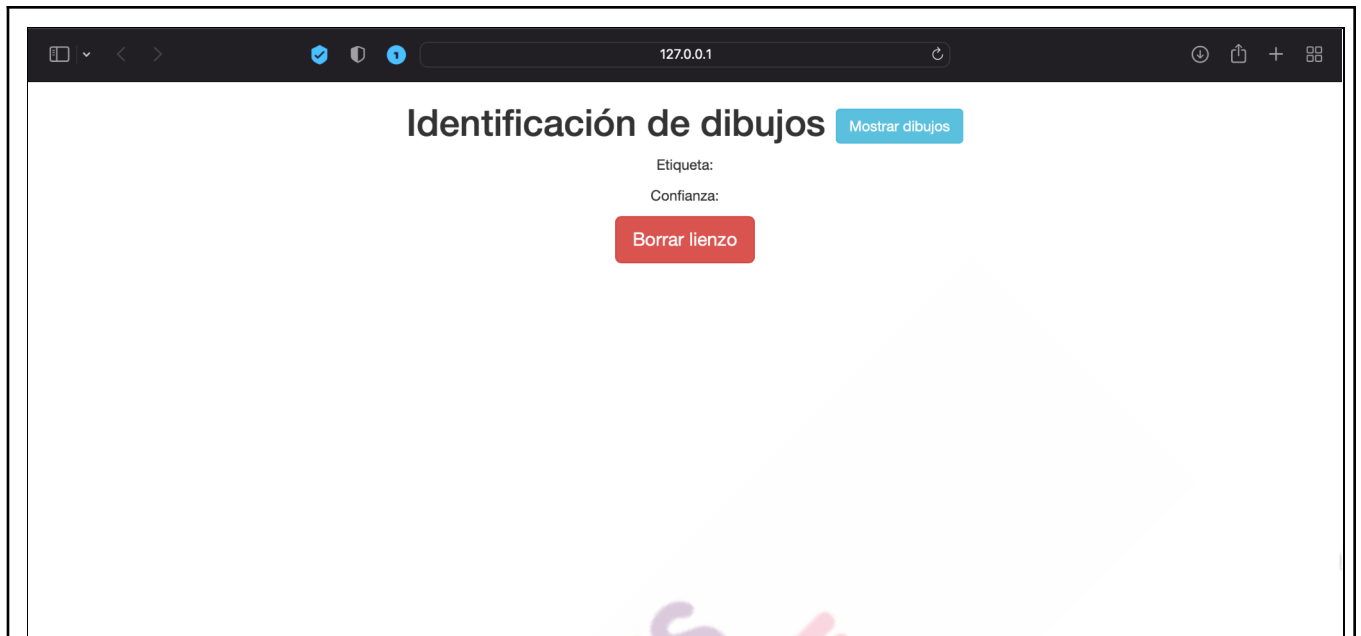
  <script src="main.js"></script>
</body>

</html>
```

En el código anterior, cubrimos los siguientes conceptos:

- Agregamos la etiqueta `<h1>` y, dentro de ella, agregamos el encabezado del sitio web.
- En la misma etiqueta `<h1>`, agregamos un botón para abrir el modal de bootstrap que muestra las imágenes de los dibujos.
- Agregamos una etiqueta `<p>` para contener la etiqueta y la confianza del dibujo identificado.
- Agregamos otro botón para borrar el output del lienzo.

Hasta ahora, nuestra aplicación se ve así:



En la clase de hoy, haremos lo siguiente:

- Programar en JS para agregar un lienzo y colocarlo en el centro.
- Agregar estilo al encabezado.
- Agregar estilo al texto de la etiqueta y confianza en la página web.
- Agregar animación al lienzo.

Código CSS completo:

```
body
{
background-color: #fbd139;
}

h1
{
color: rgb(38, 0, 255);
background-color: cornsilk;
}

#show_sketch
{
float:right;
}
```

```
canvas
{
  border:5px solid ■white;
  border-radius:20px;
  animation-name: example;
  animation-duration: 4s;
  margin-top: 50px;
  animation-iteration-count: infinite;
}
@keyframes example {
  from {box-shadow: 1px 1px 38px ■grey;}
  to {box-shadow: 1px 1px 38px ■white;}
}

p
{
  padding-top:10px;
  font-size:25px;
}
```

Código JS:

```
function setup() {  
  canvas = createCanvas(280, 280);  
  canvas.center();  
  background("white");  
}  
  
function clearCanvas() {  
  background("white");  
}
```

Comencemos a agregar estilo a los elementos que hemos creado en el archivo html:

- 1) **Establecer el color de fondo de la aplicación web:** Comencemos agregando el color de fondo a la página web. Para esto, necesitamos agregar código para establecer el **color de fondo** de la página web. Ya hemos definido la etiqueta de **body** (cuerpo) en el archivo html en la clase anterior. Accedamos a la etiqueta de **body** que se ve así:

```
body  
{  
  
}
```

Y dentro de esta etiqueta, necesitamos agregar el código para establecer el color de fondo de la página web. Agregaremos el color amarillo a la página web, así:

```
body  
{  
  background-color: #fbd139;  
}
```

Puedes agregar el color que desees para establecerlo como color de fondo.

Entonces el output debería verse así:



- 2) **Agregar color al título:** Ahora, agregaremos color al encabezado que es el texto 'Identificación de dibujos'. En el archivo html, ya agregamos el encabezado usando la etiqueta h1. Ahora, estableceremos el color de la fuente y el color de fondo del encabezado.

Entonces, el código se verá así:

```
h1
{
  color: ■ rgb(38, 0, 255);
  background-color: ■ cornsilk;
}
```

Aquí, necesitamos acceder a la etiqueta **h1** que se ve así:


```
h1
{
}
```

Y dentro de estos corchetes, necesitamos establecer el color de la fuente usando la propiedad de **color** y seleccionando el color de esta manera:

```
color: rgb(38, 0, 255);
```

Hemos establecido el color indicado con los valores **rgb**. Puedes hacerlo configurando el valor HEX o dándole el nombre de algún color directamente. Puedes agregar el color que desees.

Luego, en la siguiente línea, agregaremos el color de fondo para el encabezado, y para eso, necesitamos usar la propiedad **background-color**.

```
background-color: cornsilk;
```

Entonces, logramos el siguiente output:



- 3) **Alinear el botón 'Mostrar dibujos'**: Ahora, alinearemos el botón **Mostrar dibujos**, al lado derecho de la página web. Para esto, necesitamos agregar el siguiente código:

```
#show_sketch
{
  float:right;
}
```

Ahora, hemos creado el botón con la identificación (**id**) como **show_sketch** (mostrar dibujos). Mientras lo estilizamos usando el código CSS, accederemos agregando el signo **#** antes del nombre de la **id**.

Y dentro de los corchetes, usaremos la propiedad **float** (flotante) del elemento del botón **show_sketch**. Esto básicamente establece la alineación. Establecemos la

alineación en el lado derecho de la página web, de modo que el botón **Mostrar dibujos** aparezca en el extremo derecho de la página web. La salida se vería así:



- 4) **Aplica estilo a la etiqueta <p>:** Es hora de que le demos estilo a la etiqueta **<p>**. En la clase anterior, agregamos 2 etiquetas **<p>**: una para mantener la etiqueta y otra para mantener la confianza. Ahora, les agregaremos estilo usando la propiedad **padding-top** (espacio en la parte superior) y estableceremos el valor en **10** píxeles. Y también estableceremos la propiedad de **font-size** (tamaño de fuente) y estableceremos su valor en **25** píxeles.



- 5) **Crea un lienzo:** Ahora, vamos a crear un elemento canvas (lienzo) en el archivo **main.js**. Para esto, tienes que agregar código en el archivo **main.js** que se te proporcionó en la clase anterior.

```
function setup() {  
  canvas = createCanvas(280, 280);  
  canvas.center();  
  background("white");  
}
```

Entonces, aquí, podemos crear una función llamada **setup()**, y dentro de los corchetes, necesitamos escribir el código para crear un lienzo:

- Necesitamos usar la función **createCanvas()** (crear el lienzo) y pasar la altura (**height**) y el ancho (**width**) como **280** y **280** para declarar los parámetros a esta función, como se muestra a continuación:

```
canvas = createCanvas(280, 280);
```

- Ahora, colocaremos el lienzo en el centro. Para esto, necesitamos usar la función **center()** (centro) del objeto lienzo.

```
canvas.center();
```

- Luego, configuraremos el color de fondo del lienzo en blanco. Para esto, necesitamos usar la función **background()** (fondo), y dentro de la función, estableceremos el color a **white** (blanco), así:

```
background("white");
```

- 6) **Limpiar el lienzo:** Mientras estamos trazando dibujos en el lienzo, es posible que deseemos hacer otro dibujo después de completar uno. Para esto, necesitamos borrar el dibujo anterior y limpiar el lienzo, de modo que el lienzo esté listo para otro dibujo. Usaremos una función para esto, que se denomina función **clearCanvas()** (limpiar o borrar lienzo). Necesitamos llamar a esta función cuando se hace clic en el botón **Borrar lienzo**:

```
function clearCanvas() {  
  background("white");  
}
```

```
function clearCanvas() {  
}
```

- Entonces, escribiremos la función como
- Y dentro de él, agregaremos color de fondo al lienzo, lo que significa que, después de borrar la salida, el fondo del lienzo se vuelve a establecer en blanco.

```
background("white");
```

- 7) **Agregar estilo al lienzo:** Ahora, agregaremos estilo al lienzo. Entonces, volveremos a escribir código en el archivo **style.css**:

```
canvas  
{  
  border:5px solid white;  
  border-radius:20px;  
  animation-name: example;  
  animation-duration: 4s;  
  margin-top: 50px;  
  animation-iteration-count: infinite;  
}  
@keyframes example {  
  from {box-shadow: 1px 1px 38px grey;}  
  to {box-shadow: 1px 1px 38px white;}  
}
```

- Primero accederemos al elemento de lienzo y le agregaremos un borde, así:

```
border:5px solid white;
```

Estableceremos el **ancho del borde** en 5 píxeles, el **estilo del borde** en sólido y el **color del borde** a **white** (blanco).

→ Ahora, redondearemos las esquinas del lienzo. Para esto, necesitamos establecer la propiedad **border-radius** (radio del borde) en **20px**, así:

```
border-radius: 20px;
```

→ Luego, agregaremos algo de animación a este lienzo:

- **animation-name (nombre de la animación):** **example** (ejemplo);
Aquí, estamos definiendo el nombre de la animación como ejemplo.

```
animation-name: example;
```

En el siguiente punto, necesitamos definir este **example** donde agregaremos código de lo que será la animación.

- **animation-duration (duración de la animación):** **4s**;
Aquí, definiremos el tiempo de la animación, es decir, cuánto tiempo debe ejecutarse una animación.

```
animation-duration: 4s;
```

- **animation-iteration-count (Recuento de iteraciones de animación):** **infinito** (infinite);
Aquí, queremos que se ejecute la animación continuamente. Entonces, necesitamos especificar que el recuento de iteraciones sea **infinito**.

```
animation-iteration-count: infinite;
```

→ Además, necesitamos establecer el margen en la parte superior en **50px**.

```
margin-top: 50px;
```

Si tu lienzo aparece sobre el botón **Borrar lienzo**, aumenta el valor de **margin-top**, de modo que sea posible colocar el lienzo debajo del botón **Borrar lienzo**.

→ Ahora, agregamos la sombra del cuadro al lienzo de la siguiente manera:

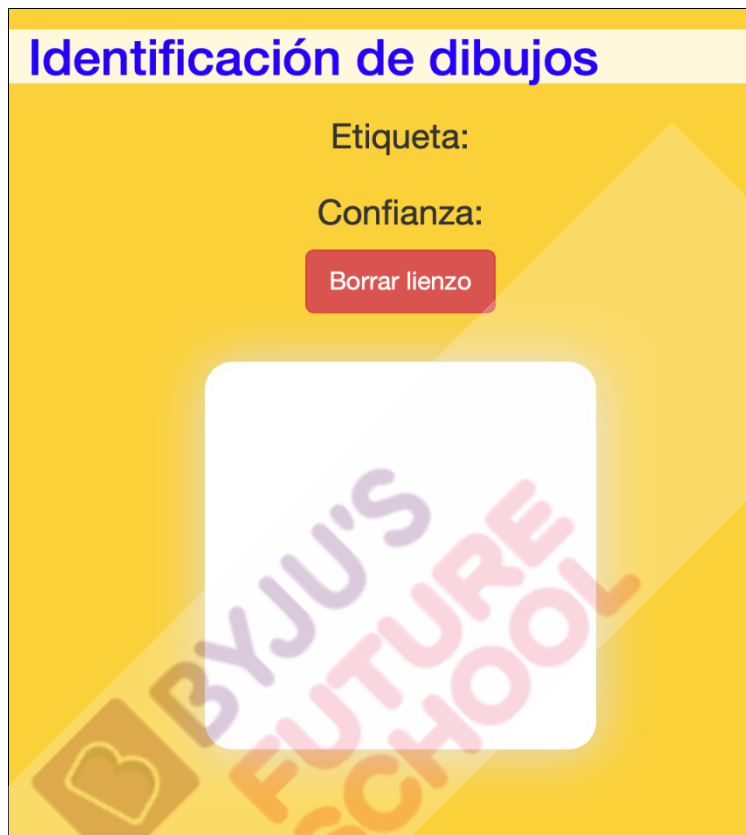
- **@keyframes example {}**
 - **@keyframes** es una palabra clave que se utiliza para definir una animación.
 - **example** es el nombre de la animación que hemos llamado en el punto anterior.
 - **from {box-shadow: 1px 1px 38px grey;}**
 - Aquí, configuramos la sombra del cuadro en gris cuando comienza la animación.
 - **to {box-shadow: 1px 1px 38px white;}**
 - Ahora, queremos que la propiedad de la sombra del cuadro cambie de **grey** (gris) a **white** (blanco). Entonces, necesitamos especificar el blanco aquí.
- Aquí, el primero **1px** significa: cuánto hacia el lado derecho del lienzo deseas el color.
 - El segundo **1px** significa: cuánto hacia la parte inferior deseas el color.
 - **38px** significa: qué tan extenso deseas que se extienda el color.

```
@keyframes example {  
  from {box-shadow: 1px 1px 38px grey;}  
  to {box-shadow: 1px 1px 38px white;}  
}
```

Básicamente, esta animación funcionará continuamente, cambiando la sombra del cuadro de gris a blanco.

Hemos usado el color gris y blanco aquí, pero puedes usar el color que desees.

OUTPUT:



La maestra deja de compartir pantalla


ACTIVIDAD DEL ALUMNO  **- 30 minutos**



Ahora es tu turno.

- **Pide al alumno que presione la tecla ESC para volver al panel.**
- **Guía al alumno para que deje de compartir la pantalla.**
- **La maestra cambia a pantalla completa.**

El alumno comienza a compartir pantalla

<p>1. Inicia la programación CSS en el archivo style.css.</p> <p>2. Inicia la programación JS en el archivo main.js.</p> <p>Mantén los archivos de código seguros, ya que cargaremos estos archivos en github cuando terminemos de construir este sitio web.</p>	<p><u>Actividad-del-alumno-1-</u> DIAGRAMA DEL CÓDIGO</p> <p><u>Actividad-de-referencia-del-alum</u> <u>no-1- REFERENCIA CSS</u> BOX-SHADOW (sombra del cuadro)</p> <p><u>Actividad-de-referencia-del-alum</u> <u>no-2- ANIMACIÓN CSS</u></p> <p>El alumno debe iniciar la programación CSS en el archivo style.css.</p> <p>El alumno debe iniciar la programación JS en el archivo main.js.</p> <p>Anima al alumno a diseñar libremente si el tiempo lo permite.</p>
<p>1) Agrega el siguiente código CSS:</p>	


```
body
{
background-color:  #fbd139;
}

h1
{
color:  rgb(38, 0, 255);
background-color:  cornsilk;
}

#show_sketch
{
float:right;
}
```

```
canvas
{
  border:5px solid white;
  border-radius:20px;
  animation-name: example;
  animation-duration: 4s;
  margin-top: 50px;
  animation-iteration-count: infinite;
}
@keyframes example {
  from {box-shadow: 1px 1px 38px grey;}
  to {box-shadow: 1px 1px 38px white;}
}

p
{
  padding-top:10px;
  font-size:25px;
}
```

2) Agrega el siguiente código JS:

```
function setup() {
  canvas = createCanvas(280, 280);
  canvas.center();
  background("white");
}

function clearCanvas() {
  background("white");
}
```

Guía al alumno para que deje de compartir pantalla

La maestra comienza a compartir pantalla

CONCLUSIÓN - 5 minutos



La maestra inicia la presentación desde la diapositiva 13 a la 16.
Consultas las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

PRÁCTICAS Y SUGERENCIAS PARA EL PROYECTO - 5 minutos



La maestra inicia la presentación desde la diapositiva 17.
Consultas las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Decir



Hacer



<p>APLICACIÓN WEB DE DIBUJO RÁPIDO - 2</p> <p>Objetivo del proyecto:</p> <p>Hoy, has terminado de diseñar una aplicación web y has dibujado un lienzo en la página web.</p> <p>En este proyecto, tendrás que completar el diseño de una página web y agregar el lienzo y el código del temporizador, usando código JS.</p> <p>Historia:</p> <p>SWork está lanzando una aplicación de dibujo para niños pequeños de 5 a 8 años, y ahora quieren una aplicación, que será un juego de dibujos, y les gustó tu interfaz de usuario. Ahora, quieren que agregues el componente de temporizador a la aplicación web de tal manera que el usuario tenga un límite de tiempo para realizar la tarea de dibujo.</p> <p>Así que comencemos agregando el estilo a la aplicación web.</p> <p>¡Buena suerte!</p>		<p>Abre el enlace de la solución del proyecto y muestra el proyecto al alumno.</p>
ACTIVIDAD ADICIONAL		
La maestra comienza a compartir pantalla		
<p>La maestra inicia la presentación  desde la diapositiva 18 a la 22.</p>		

ACTIVIDAD ADICIONAL DEL ALUMNO



El alumno comienza a compartir pantalla

NOTA PARA LAS MAESTRAS -

Para la solución de todas las actividades adicionales abre **actividad-de-la-maestra-4** y navega por el número de clase **C118**.

Actividad adicional 1 -

Ejecuta la **actividad-del-alumno-2** desde el **panel**.

La **TAREA** y **SUGERENCIAS** se mencionan en el sitio web.

Actividad adicional 2 -

Ejecuta la **actividad-del-alumno-3** desde el **panel**.

La **TAREA** y **SUGERENCIAS** se mencionan en el sitio web.

Actividad adicional 3 -

Ejecuta la **actividad-del-alumno-4** desde el **panel**.

La **TAREA** y **SUGERENCIAS** se mencionan en el sitio web.

Actividad adicional 4 -


Ejecuta la **actividad-del-alumno-5** desde el **panel**.

La **TAREA** y **SUGERENCIAS** se mencionan en el sitio web.

Actividad adicional 5 -

Ejecuta la **actividad-del-alumno-6** desde el **panel**.

La **TAREA** y **SUGERENCIAS** se mencionan en el sitio web.

La maestra finaliza la presentación  **en la diapositiva 23.**
Consulta las notas en tu pantalla y sigue las instrucciones

La maestra deja de compartir pantalla

La maestra hace clic en

✕ Finalizar Clase

Actividad	Nombre de la actividad	Enlaces
Actividad de la maestra 1	DIAGRAMA DEL CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/c54158a8-4691-43e0-83e5-9a966a1dfbc7.pdf
Actividad de la maestra 2	DEMOSTRACIÓN BOX SHADOW	https://developer.mozilla.org/es/docs/Web/CSS/box-shadow
Actividad de la maestra 3	CODIGO COMPLETO	https://drive.google.com/drive/folders/1qs1XxXvdsrx-8viVEOgc3CfJA_yMhAyu?usp=sharing
Actividad de la maestra 4	SOLUCIONES DE LAS ACTIVIDADES ADICIONALES	https://docs.google.com/spreadsheets/d/e/2PACX-1vSkD-LoPgaAem9vlwHH_AjcByD8RX0nmgSwpS8EHtO1jJzSs4ECJTckimy8o4OmaScgcl5U_kaSaNoC/pubhtml
Actividad del alumno 1	DIAGRAMA DEL CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/c54158a8-4691-43e0-83e5-9a966a1dfbc7.pdf
Actividad de referencia para el alumno 1	REFERENCIA CSS BOX-SHADOW	https://developer.mozilla.org/es/docs/Web/CSS/box-shadow
Actividad de referencia para el alumno 2	ANIMACIÓN CSS	https://developer.mozilla.org/es/docs/Web/CSS/animation
Solución del proyecto	APLICACIÓN WEB DE DIBUJO RÁPIDO - 2	https://drive.google.com/file/d/1xYvVeDQB501L8x3m-Mb3-PWnzT6PxQhN/view?usp=sharing

Apoyo visual de referencia para la maestra	Enlace del apoyo visual con notas	https://s3-whjr-curriculum-uploads.whjr.online/585f0977-9f20-4a83-b1f0-af16e2142d06.html
Quiz de la clase de referencia para la maestra	Quiz de la clase	https://s3-whjr-curriculum-uploads.whjr.online/1ab01e23-40f1-4ded-9f32-109d0a12a2d1.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/0c698abb-aaa4-4c0d-bb29-ba8fb3971c2b.html
--------------------------	------------------------	---