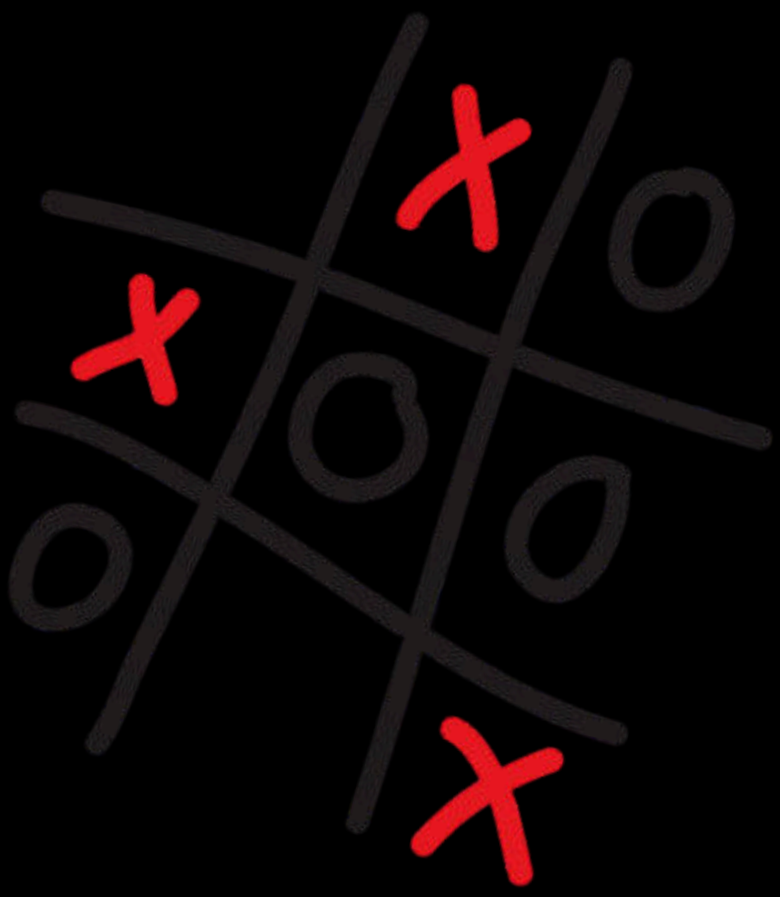
tictactoe



Manual y juego elaborado por:

Evelyn Paez

DIseño y maquetacion

Rodrigo Alejandro

🎮 MANUAL DE USO DEL JUEGO “TIC TAC TOE”

🟦 Introducción

El juego Tic Tac Toe (también conocido como “Gato” o “Tres en línea”) es una versión digital del clásico juego de mesa donde dos jugadores, X y O, compiten por alinear tres símbolos iguales de manera horizontal, vertical o diagonal.
Este proyecto fue desarrollado utilizando HTML, CSS y JavaScript, tres tecnologías esenciales para crear páginas web interactivas:

- HTML define la estructura visual del tablero y los elementos del juego. CSS da estilo y diseño, mejorando la
- presentación visual. JavaScript controla la lógica del juego: los turnos, las condiciones de victoria
- y la reinicialización.

🎮 MANUAL DE USO DEL JUEGO “TIC TAC TOE”

🟦 Introducción

El juego Tic Tac Toe (también conocido como “Gato” o “Tres en línea”) es una versión digital del clásico juego de mesa donde dos jugadores, X y O, compiten por alinear tres símbolos iguales de manera horizontal, vertical o diagonal.
Este proyecto fue desarrollado utilizando HTML, CSS y JavaScript, tres tecnologías esenciales para crear páginas web interactivas:

- HTML define la estructura visual del tablero y los elementos del juego. CSS da estilo y diseño, mejorando la
- presentación visual. JavaScript controla la lógica del juego: los turnos, las condiciones de victoria
- y la reinicialización.

1. El proyecto se compone de tres archivos principales:
2. tic.html – Estructura y contenido del juego.
3. tic.css – Apariencia, colores y estilos visuales.
4. tic.js – Programación y funcionamiento del juego.

🧱 1. Documento HTML — tic.html

Este archivo define la estructura visual de todo el juego.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tic_Tac_Toe</title>
    <link rel="stylesheet" href="tic.css">
</head>
```

- <html lang="en">: define el idioma del documento (inglés). <meta
- charset="UTF-8">: permite usar caracteres especiales. <meta
- name="viewport">: hace que el juego sea adaptable a celulares o tablets. <title>: título que aparece en la
- pestaña del navegador. <link rel="stylesheet" href="tic.css">:
- enlaza el archivo de estilos CSS.

Cuerpo del documento

```
<body>
    <div class="turn-container">
        <h3>Turn For</h3>
        <div class="turn-box align">X</div>
        <div class="turn-box align">O</div>
        <div class="bg"></div>
    </div>
```

- <html lang="en">: define el idioma del documento (inglés). Muestra un **indicador de turno** para saber si le
- toca jugar a “X” o “O”. El elemento .bg (de color rosa) se mueve de un
- lado al otro, marcando quién juega.

◆ Tablero principal del juego

```
<div class="main-grid">
    <div class="box align">0</div>
    <div class="box align">1</div>
    <div class="box align">2</div>
    <div class="box align">3</div>
    <div class="box align">4</div>
    <div class="box align">5</div>
    <div class="box align">6</div>
    <div class="box align">7</div>
    <div class="box align">8</div>
</div>
```

- Es la **rejilla de 3x3** donde se juega.
- Cada representa una celda (del 0 al 8).
- El número es solo una referencia; el jugador lo reemplaza con una **X** o **O** al hacer clic.

🔹 Sección de resultados y botón

```
<h2 id="results"></h2>
<button id="play-again">Play Again</button>
```

- **#results** muestra el resultado final (“X win”, “O win” o “Draw”).
- **#play-again** reinicia el juego cuando termina.

🔹 Conexión con el script

```
<script src="tic.js"></script>
</body>
</html>
```

🎮 Manual del código tic.css Juego:

Tic Tac Toe (Tres en Raya) **Archivo:** tic.css **Función:** Este documento define el **aspecto visual (diseño, colores, distribución y animaciones)** del juego. Controla cómo se ven los cuadros, los textos, el tablero y los botones.

🔹 Estilos generales

```css
*{
    color: white;
    font-family: sans-serif;
    transition: 0.2s ease-in-out;
    user-select: none;
}
```

- * → Afecta **a todos los elementos** del documento. color: white; → Define el **color del texto blanco**
- por defecto. font-family: sans-serif; → Usa una **fuente sin remates**, moderna y legible.
- transition: 0.2s ease-in-out; → Aplica una **transición suave** (0.2 segundos) a los cambios visuales como color o tamaño. user-select:
- none; → Evita que el jugador seleccione texto accidentalmente durante el juego.

-

🔹 Alineación central común

```
.align{
    display: flex;
    justify-content: center;
    align-items: center;
}
```

- .align es una clase reutilizable que centra su contenido tanto **horizontal como verticalmente.** display: flex; → Convierte al
- elemento en un **contenedor flexible**. justify-content: center; y align- items: center; → Centran el texto o contenido
- dentro del cuadro.

Se usa para centrar las X y O dentro de cada casilla del tablero.

- Estilos del cuerpo del juego

```
body{
    background-color: #252A34;
    margin: 0;
    padding: 0;
    width: 100vw;
    text-align: center;
    padding-top: 5vh;
}
```

- background-color: #252A34; → Fija un **fondo oscuro azul-grisáceo**, que da contraste con el texto blanco. margin y padding: 0; → Elimina los
- espacios por defecto del navegador. width: 100vw; → Usa todo el ancho de la pantalla (“viewport width”). text-align:
- center; → Centra todos los textos dentro del cuerpo. padding-top: 5vh; → Deja espacio en la parte
- superior (5% del alto de la ventana).
-

🔹 Contenedor del turno del jugador

```css
.turn-container{
    width: 170px;
    height: 80px;
    margin: auto;
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: 1fr 1fr;
    position: relative;
}
```

- display: grid; → Crea una **rejilla** con dos columnas (una para X y otra para O).
- grid-template-columns: 1fr 1fr; → Divide el espacio en dos partes iguales.
- position: relative; → Permite colocar un fondo (.bg) encima de este contenedor.
- width y height → Determinan el tamaño del cuadro superior.
- margin: auto; → Lo centra horizontalmente en la pantalla.
-

🔹 Título “Turn For”

```
.turn-container h3{
    margin: 0;
    grid-column-start: 1;
    grid-column-end: 3;
}
```

- Aplica estilo al texto "Turn For".
- grid-column-start y grid-column-end → Hacen que el título ocupe dos columnas completas del contenedor.
- margin: 0; → Elimina márgenes por defecto para mantener el espacio uniforme.

◆ Cuadros que muestran X y O

```
.turn-container .turn-box{
    border: 3px solid #000;
    font-size: 1.6rem;
    font-weight: 700;
}
```

- Crea los **cuadros donde aparecen X y O**.
- border: 3px solid #000; → Les da un **borde negro grueso**.

- font-size y font-weight → Hacen que las letras X y O sean grandes y visibles.

```
.turn-container .turn-box:nth-child(even){
    border-right: none;
}
```

- :nth-child(even) → Selecciona el segundo cuadro (O).
- border-right: none; → Elimina el borde entre ambos cuadros, para que se vean como un solo bloque dividido.

Fondo animado del turno

```
.bg{
    position: absolute;
    bottom: 0;
    left: 0;
    width: 85px;
    height: 40px;
    background-color: #FF2E63;
    z-index: -1;
}
```

- .bg representa una barra de color que se mueve según el turno. position: absolute; → Permite
- colocarla sobre los cuadros. left: 0; → Inicia debajo de la "X".
- En el JavaScript, se cambia su posición (left: 85px;) para moverse al lado de la "O".
- z-index: -1; → La coloca detrás del texto para que no lo tape. background-color: #FF2E63; → Color rosa-rojizo brillante que
- destaca el turno actual.

◆ Tablero principal

```css
.main-grid{
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(3, 1fr);
    height: 250px;
    width: 250px;
    margin: 30px auto;
    border: 2px solid #000;
}
```

- Define la rejilla del juego (3x3).
- grid-template-columns: repeat(3, 1fr); → Tres columnas iguales.
- grid-template-rows: repeat(3, 1fr); → Tres filas iguales.
- height y width → El tablero mide 250px de lado.
- margin: 30px auto; → Lo centra y deja un espacio superior.
- border → Agrega un borde negro alrededor del tablero.

◆ Casillas individuales

```css
.box{
    cursor: pointer;
    font-size: 2rem;
    font-weight: 700;
    border: 2px solid #000;
}
```

- .box son las nueve casillas del tablero.
  cursor: pointer; → Cambia el
- cursor a una mano al pasar sobre la casilla.
  font-size → Tamaño grande de texto para X y O.
- font-weight → Negrita.
  border → Define los bordes entre casillas, simulando la cuadrícula
- del tres en raya.
-

Efecto al pasar el mouse

```
.box:hover{
    background-color: #FF2E63;
}
```

Agranda el botón ligeramente y cambia su color a **turquesa (#08D9D6)** cuando se pasa el cursor.
color: #000; → Cambia el texto a negro para mantener contraste.

⚙️ 3. Archivo JavaScript – *tic.js*

Aquí está toda la **lógica y funcionamiento del juego**.

🔹 Variables iniciales

```
let boxes = document.querySelectorAll(".box");
let turn = "X";
let isGameOver = false;
```

- boxes: contiene las 9 casillas del tablero.
- turn: almacena de quién es el turno actual ("X" o "O").
- isGameOver: indica si la partida terminó.
- Detección de clics en cada casilla y

🔹

```
boxes.forEach(e =>{
    e.innerHTML = ""
    e.addEventListener("click", ()=>{
        if(!isGameOver && e.innerHTML === ""){
            e.innerHTML = turn;
            cheakWin();
            cheakDraw();
            changeTurn();
        }
    })
})
```

- Limpia las casillas al iniciar. Cuando se hace clic en una celda
- vacía:
    - Coloca el símbolo del turno actual.
    - Verifica si alguien ganó o si hay empate.
    - Cambia el turno.

🔹 Cambio de turno

```
function changeTurn(){
    if(turn === "X"){
        turn = "O";
        document.querySelector(".bg").style.left = "85px";
    }
    else{
        turn = "X";
        document.querySelector(".bg").style.left = "0";
    }
}
```

- Alterna entre “X” y “O”.
  Mueve el fondo rosado (.bg) debajo
- del jugador actual.

🔹 Comprobación de victoria

```
function cheakWin(){
    let winConditions = [
        [0,1,2],[3,4,5],[6,7,8],
        [0,3,6],[1,4,7],[2,5,8],
        [0,4,8],[2,4,6]
    ];
```

- Estas combinaciones son las **8 formas posibles de ganar**.

Si una línea tiene tres símbolos iguales:

```
isGameOver = true;
document.querySelector("#results").innerHTML = turn + " win";
document.querySelector("#play-again").style.display = "inline";
```

- Muestra el ganador.
  Activa el botón “Play Again”.
- Cambia el color de las casillas ganadoras a celeste.
-

◆ Comprobación de empate

```
function cheakDraw(){
    if(!isGameOver){
        let isDraw = true;
        boxes.forEach(e =>{
            if(e.innerHTML === "") isDraw = false;
        })
        if(isDraw){
            isGameOver = true;
            document.querySelector("#results").innerHTML = "Draw";
            document.querySelector("#play-again").style.display = "inline";
        }
    }
}
```

- Si todas las casillas están llenas y nadie ganó → empate.

◆ **Reinicio del juego**

```
document.querySelector("#play-again").addEventListener("click", ()=>{
    isGameOver = false;
    turn = "X";
    document.querySelector(".bg").style.left = "0";
    document.querySelector("#results").innerHTML = "";
    document.querySelector("#play-again").style.display = "none";

    boxes.forEach(e =>{
        e.innerHTML = "";
        e.style.removeProperty("background-color");
        e.style.color = "#fff";
    })
})
```

- Restablece todos los valores. Vacía las casillas.
- Devuelve el turno al jugador “X”. Oculta el botón hasta la siguiente
- partida.
-

🧠 Conclusión

El juego **Tic Tac Toe** demuestra cómo combinar las tres tecnologías básicas del desarrollo web:

- **HTML** crea la estructura del tablero y los elementos del juego.
  CSS define su apariencia moderna y
- atractiva.
  JavaScript agrega la inteligencia del juego, controlando los turnos,
- victorias y reinicio.

Gracias a su simplicidad, este proyecto es ideal para aprender los fundamentos de la **interactividad web** y la manipulación del **DOM**.
Además, el diseño es completamente adaptable y visualmente agradable para jugar tanto en computadora como en dispositivos móviles.