# Tic tac toe

## Nombre:

Tic tac toe

## ¿Como se realizó?

Primero previsualice lo que quería hacer en un algoritmo u seudocódigo,

Posterior a ello comencé a primeramente realizar el cuerpo en HTML, y luego a ello procedí a darle un diseño un poco minimalista con css Grid, ya por ultimo le di vida al juego utilizando JavaScript.

**Código o maquetación utilizado en HTML:**

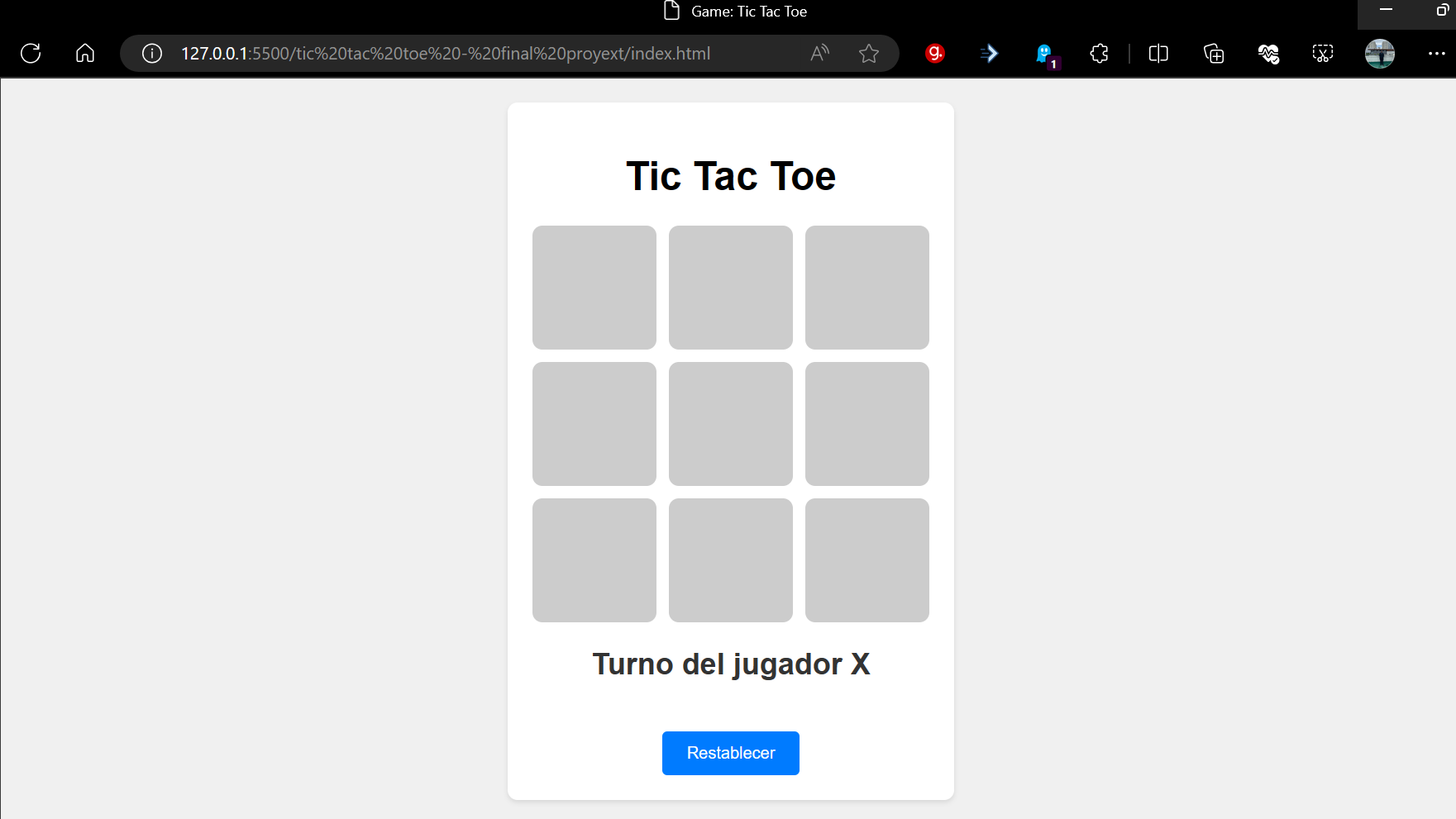
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | <!doctype html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Game: Tic Tac Toe</title>  <link rel="stylesheet" href="style.css">  </head>  <body>  <div class="container">  <h1>Tic Tac Toe</h1>  <div class="game-container">  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  <div class="game-cell"></div>  </div>  <h2 class="game-notification"></h2>  <button class="game-restart">Restablecer</button>  </div>  <script src="main.js"></script>  </body>  </html> |

**Estilo utilizado con CSS:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56 | body {  **font-family**: Arial, **sans-serif**;  **margin**: **0**;  **padding**: **0**;  **display**: flex;  **justify**-**content**: **center**;  align-items: **center**;  **height**: **100**vh;  **background-color**: **#f0f0f0**;  }  **.container** {  **text-align**: **center**;  **background-color**: **#fff**;  **padding**: **20px**;  **border**-radius: **8px**;  box-shadow: **0** **2px** **4px** rgba(**0**, **0**, **0**, **0**.**1**);  }  **.game-container** {  **display**: grid;  grid-template-columns: **repeat**(**3**, **100px**);  grid-gap: **10px**;  **margin-top**: **20px**;  }  **.game-cell** {  **width**: **100px**;  **height**: **100px**;  **background-color**: **#ccc**;  **border**-radius: **8px**;  **display**: flex;  **justify**-**content**: **center**;  align-items: **center**;  **font-size**: **2em**;  }  **.game-notification** {  **margin-top**: **20px**;  **color**: **#333**;  }  **.game-restart** {  **margin-top**: **20px**;  **padding**: **10px** **20px**;  **background-color**: **#007bff**;  **color**: **#fff**;  **border**: **none**;  **border**-radius: **4px**;  **cursor**: **pointer**;  transition: **background-color** **0.3s** ease;  }  **.game-restart:hover** {  **background-color**: **#0056b3**;  } |

**Código utilizado con JS para darle vida al juego, documentación agregada:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109 | // ==================== CONSTANTE ==================== //  **const** STATUS\_DISPLAY = document.querySelector('.game-notification'),  GAME\_STATE = ["", "", "", "", "", "", "", "", ""],  WINNINGS = [  [**0**, **1**, **2**],  [**3**, **4**, **5**],  [**6**, **7**, **8**],  [**0**, **3**, **6**],  [**1**, **4**, **7**],  [**2**, **5**, **8**],  [**0**, **4**, **8**],  [**2**, **4**, **6**]  ],  WIN\_MESSAGE = () => `El jugador ${currentPlayer} ha ganado!`,  DRAW\_MESSAGE = () => `El juego ha terminado en empate!`,  CURRENT\_PLAYER\_TURN = () => `Turno del jugador ${currentPlayer}`  // ==================== VARIABLES ==================== //  **let** gameActive = **true**,  currentPlayer = "O"  // ==================== FUNCTIONS ==================== //  **function** main() {  handleStatusDisplay(CURRENT\_PLAYER\_TURN())  listeners()  }  **function** listeners() {  document.querySelector('.game-container').addEventListener('click', handleCellClick)  document.querySelector('.game-restart').addEventListener('click', handleRestartGame)  }  **function** handleStatusDisplay(message) {  STATUS\_DISPLAY.innerHTML = message  }  **function** handleRestartGame() {  gameActive = **true**  currentPlayer = "X"  restartGameState()  handleStatusDisplay(CURRENT\_PLAYER\_TURN())  document.querySelectorAll('.game-cell').forEach(cell => cell.innerHTML = "")  }  **function** handleCellClick(clickedCellEvent /\*\* Type Event \*\*/) {  **const** clickedCell = clickedCellEvent.target  **if** (clickedCell.classList.contains('game-cell')) {  **const** clickedCellIndex = Array.from(clickedCell.parentNode.children).indexOf(clickedCell)  **if** (GAME\_STATE[clickedCellIndex] !== '' || !gameActive) {  **return** **false**  }  handleCellPlayed(clickedCell, clickedCellIndex)  handleResultValidation()  }  }  **function** handleCellPlayed(clickedCell /\*\* object HTML \*\*/, clickedCellIndex) {  GAME\_STATE[clickedCellIndex] = currentPlayer // Agrega en la posición correspondiente el valor ya sea "X" u "O" en el estado actual del juego  clickedCell.innerHTML = currentPlayer // Agrega en el HTML el valor del jugador  }  **function** handleResultValidation() {  **let** roundWon = **false**  **for** (**let** i = **0**; i < WINNINGS.length; i++) { // Itera cada uno de las posibles combinaciones ganadores  **const** winCondition = WINNINGS[i] // Guarda la combinación por ejemplo: [0, 1, 2]  **let** position1 = GAME\_STATE[winCondition[**0**]],  position2 = GAME\_STATE[winCondition[**1**]],  position3 = GAME\_STATE[winCondition[**2**]] // Almacena el valor del estado actual del juego según las posiciones de winCondition  **if** (position1 === '' || position2 === '' || position3 === '') {  **continue**; // Si hay algún valor vacio nadie ha ganado aún  }  **if** (position1 === position2 && position2 === position3) {  roundWon = **true** // Si todas las posiciones coinciden entonces, dicho jugador ha ganado la partida  **break**  }  }  **if** (roundWon) {  handleStatusDisplay(WIN\_MESSAGE())  gameActive = **false**  **return**  }  **let** roundDraw = !GAME\_STATE.includes("") // Si todas las celdas tienen valor y la sentencia anterior fue falsa entonces es empate  **if** (roundDraw) {  handleStatusDisplay(DRAW\_MESSAGE())  gameActive = **false**  **return**  }  handlePlayerChange()  }  **function** handlePlayerChange() {  currentPlayer = currentPlayer === "X" ? "O" : "X"  handleStatusDisplay(CURRENT\_PLAYER\_TURN())  }  **function** restartGameState() {  **let** i = GAME\_STATE.length  **while** (i--) {  GAME\_STATE[i] = ''  }  }  main() |

**ANEXOS:  
**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**