Reporte Segundo Proyecto: Juego de Totito

1st Annelís Juany Sacalxot Chojolán
Facultad de Ingeniería
Universidad Mesoamericana
Quetzaltenango, Guatemala
Sección E Carnet 202508040
annelisjuany_sacalxotchojolan@umes.edu.gt

2nd Sergio Alejandro Sagastume González
Facultad de Ingeniería
Universidad Mesoamericana
Quetzaltenango, Guatemala
Sección E Carnet 202508005
sergiosagastume4@umes.edu.gt

***Abstract*—Este proyecto consiste en el desarrollo de una versión minimalista del clásico juego "totito" a través de una matriz de 3x3, ejecutado en el lenguaje C. El cual incluye tres distintos tipos de nivel siendo estos: fácil, medio y difícil. La implementación utiliza funciones estructuradas, manipulación de gráficos en consola y detección automática de jugadas ganadoras. El juego tiene una modalidad de usuario contra CPU, donde el usuario a través de las teclas W, A, S, D pueden ubicarse en distintos puntos de manera interactiva. El juego proporciona una experiencia dinámica y visual a través de la consola, integrando lógica condicional, ciclos, vectores bidimensionales y control de flujo.**

I. INTRODUCCIÓN

Este proyecto consiste en la creación de una versión minimalista del juego Totito (también llamado tres en línea), programado en el lenguaje C y ejecutado desde la consola de Windows. El objetivo principal es permitir que el usuario juegue contra la computadora usando un tablero de 3x3, en el que ambos compiten para alinear tres símbolos iguales (X o O) en fila, columna o diagonal.

El juego tiene tres niveles de dificultad: fácil, medio y difícil. En el nivel fácil, la computadora juega de forma aleatoria sin "atacar al adversario"; en el medio, puede bloquear al jugador y anticipar las jugadas; y en el difícil, se tratá de bloquear todas las jugadas del usuario, haciendo que siempre gane o empate la computadora. Los movimientos del jugador se realizan con las teclas W, A, S y D, para moverse por el tablero, y con la tecla ENTER para colocar su jugada.

Durante el desarrollo se utilizaron vectores bidimensionales para representar el tablero, funciones para organizar el código, estructuras condicionales, ciclos y colores en consola para hacer más visual el juego. Este proyecto ayudó a reforzar los conocimientos básicos de programación en C, como el uso de variables, control de flujo, y manejo del teclado en tiempo real y funciones especificas como el gotoxy() y setConsoleCursorPosition().

II. DISEÑO DEL JUEGO

El diseño del juego se basa en un recuadro de 20 pixeles de largo y 20 pixeles de ancho, el cual, al jugar se divide en 9 recuadros de igual tamaño. El menú de juego le permite al jugador elegir la dificultad en la cual desea enfrentar a la máquina, siendo las opciones disponibles: FACIL, MEDIO y DIFICIL. Después de seleccionar la dificultad, el programa se encarga de decidir de manera aleatoria a quien le corresponde el primer turno, si a la maquina o al jugador, asimismo es aleatoria la asignación del símbolo a utilizar, están disponibles la X (en color azul) y las O (en color rojo).

El juego se basa en las reglas generales del totito, gana quien logre colocar 3 símbolos iguales, ya sea de manera horizontal, vertical o diagonal, al final la jugada ganadora se resaltará en un color verde para su fácil observación. Si la cuadricula de 3x3 recuadros se llena sin que se consigan los 3 símbolos seguidos, esto se tomará como un empate.

III. DESARROLLO E IMPLEMENTACIÓN DETALLADA DEL CODIGO EN C

1) Librerías utilizadas:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

#include <conio.h>

#include <ctype.h>
```

2) Definición de las teclas que el usuario va a poder utilizar durante el juego:

```
#define ARRIBA 'w'

#define ABAJO  's'

#define IZQUIERDA 'a'

#define DERECHA 'd'

#define ENTER 13
```

3) Prototipos de las funciones:

```
int gotoxy(int x, int y);

void dibujarMarco();

void pantallaCarga();

void mostrarMensajeFinal(const char* mensaje);

void dibujarTablero(char tablero[3][3]);

int verificarGanador(char tablero[3][3], char jugador);

void cpuMovimientoMedio(char tablero[3][3], char jugador, char cpu);

void cpuMovimientoDificil(char tablero[3][3], char jugador, char cpu);

void jugarFacil();

void jugarMedio();

void jugarDificil();

void mostrarSalida();

void menuPrincipal();

void menuInicio();

void resaltarJugadaGanadora(char tablero[3][3], char jugador);
```

4) Definición de los colores para los símbolos X y O:

```
#define COLOR_X 12

#define COLOR_O 9
```

5) Función para mover el cursor a una posición específica en la consola:

```
int gotoxy(int x, int y) {
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
    return 1;
}
```

6) Función para cambiar el color del texto:

```
void cambiarColor(int color) {
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), color);
}
```

7) Función para dibujar el recuadro general que se usa en menú y juego:

```
void dibujarMarco() {
    int posX, posY;
    gotoxy(4, 4);   printf("%c", char(201));
    gotoxy(40, 4);  printf("%c", char(187));
    gotoxy(4, 24);  printf("%c", char(200));
    gotoxy(40, 24); printf("%c", char(188));
    for (posX = 5; posX < 40; posX++) {
        gotoxy(posX, 4);  printf("%c", char(205));
        gotoxy(posX, 24); printf("%c", char(205));
    }
    for (posY = 5; posY < 24; posY++) {
        gotoxy(4, posY);  printf("%c", char(186));
        gotoxy(40, posY); printf("%c", char(186));
    }
}
```

8) Función de la pantalla de carga del juego

```
void pantallaCarga() {
    system("cls");
    dibujarMarco();
    for (int i = 0; i < 6; i++) {
```

```
            gotoxy(14, 14);
            if (i % 2 == 0) cambiarColor(COLOR_X);
            else cambiarColor(COLOR_O);
            printf("CARGANDO TOTITO...");
            Sleep(300);
        }
        cambiarColor(7);
}
```

9) Función para imprimir el resultado de la partida al jugador (gano, empato perdió):

```
void mostrarMensajeFinal(const char* mensaje) {
    system("cls");
    dibujarMarco();
    gotoxy(18, 14);
    printf("%s", mensaje);
    gotoxy(0, 25);
    Sleep(5000);
}
```

10) Función que dibuja el tablero de juego y coloca los símbolos:

```
void dibujarTablero(char tablero[3][3]) {
    int baseX = 5, baseY = 6;
    for (int i = 0; i < 2; i++) {
            int y = baseY + 6 * (i + 1);
            for (int x = 5; x < 40; x++) {
                gotoxy(x, y);
                printf("*");
            }
    }
    for (int i = 0; i < 2; i++) {
            int x = baseX + 11 * (i + 1);
            for (int y = 5; y < 24; y++) {
                gotoxy(x, y);
                printf("*");
            }
    }

    for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                int x = baseX + j * 11 + 4;
                int y = baseY + i * 6 + 2;
                gotoxy(x, y);
                if (tablero[i][j] == 'X') {
    cambiarColor(COLOR_X);
                } else if (tablero[i][j] == 'O') {
    cambiarColor(COLOR_O);
                }
                printf("%c", tablero[i][j]);
                cambiarColor(7);
            }
    }
}
```

11) Función para detectar la fila de símbolos ganadores horizontal, vertical o diagonal:

```
int verificarGanador(char tablero[3][3], char jugador) {
    for (int i = 0; i < 3; i++) {
            if (tablero[i][0] == jugador && tablero[i][1] == jugador && tablero[i][2] == jugador) return 1;
            if (tablero[0][i] == jugador && tablero[1][i] == jugador && tablero[2][i] == jugador) return 1;
    }
```

```
    if (tablero[0][0] == jugador && tablero[1][1] == jugador && tablero[2][2] == jugador) return 1;
    if (tablero[0][2] == jugador && tablero[1][1] == jugador && tablero[2][0] == jugador) return 1;
    return 0;
}
```

12) Función para verificar si el tablero de tres en línea está completamente lleno:

```
int tableroLleno(char tablero[3][3]) {
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            if (tablero[i][j] == ' ') return 0;
    return 1;
}
```

13) Funciones de la CPU para elegir movimientos en niveles Medio y Difícil:

```
void cpuMovimientoMedio(char tablero[3][3], char jugador, char cpu) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (tablero[i][j] == ' ') {
                tablero[i][j] = jugador;
                if (verificarGanador(tablero, jugador)) {
                    tablero[i][j] = cpu;
                    return;
                }
                tablero[i][j] = ' ';
            }
        }
    }

    int r, c;
    do {
        r = rand() % 3;
        c = rand() % 3;
    } while (tablero[r][c] != ' ');
    tablero[r][c] = cpu;
}

void cpuMovimientoDificil(char tablero[3][3], char jugador, char cpu) {

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (tablero[i][j] == ' ') {
                tablero[i][j] = cpu;
                if (verificarGanador(tablero, cpu)) {
                    return;
                }
                tablero[i][j] = ' ';
            }
        }
    }

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (tablero[i][j] == ' ') {
                tablero[i][j] = jugador;
                if (verificarGanador(tablero, jugador)) {
                    tablero[i][j] = cpu;
```

```
                                return;
                        }
                        tablero[i][j] = ' ';
                }
        }
    }


    int r, c;
    do {
        r = rand() % 3;
        c = rand() % 3;
    } while (tablero[r][c] != ' ');
    tablero[r][c] = cpu;
}
```

14) Funciones ara controlar las dificultades, según elija el jugador.

```
void jugarFacil() {
    pantallaCarga();
    char tablero[3][3] = { {' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', '
'} };
    int fila = 0, col = 0, jugadas = 0;
    char jugador, cpu;
    jugador = rand() % 2 == 0 ? 'X' : 'O';
    cpu = jugador == 'X' ? 'O' : 'X';


    if (rand() % 2 == 1) {
        int r, c;
        do {
            r = rand() % 3;
            c = rand() % 3;
        } while (tablero[r][c] != ' ');
        tablero[r][c] = cpu;
        jugadas++;
    }


    while (1) {
        system("cls");
        dibujarMarco();
        dibujarTablero(tablero);
        int x = 5 + col * 11 + 4;
        int y = 6 + fila * 6 + 2;
        gotoxy(x, y);


        char tecla = toupper(getch());
        if (tecla == 'W' && fila > 0) fila--;
        else if (tecla == 'S' && fila < 2) fila++;
        else if (tecla == 'A' && col > 0) col--;
        else if (tecla == 'D' && col < 2) col++;
        else if (tecla == '\r') {
            if (tablero[fila][col] == ' ') {
                tablero[fila][col] = jugador;
                jugadas++;


                if (verificarGanador(tablero, jugador)) {
                    system("cls");

    dibujarMarco();

    dibujarTablero(tablero);

    resaltarJugadaGanadora(tablero, jugador);
```

```
            mostrarMensajeFinal("GANO");
                                    return;
                                }


                                if (jugadas == 9) {
                                    system("cls");

            dibujarMarco();

            dibujarTablero(tablero);
                                    Sleep(5000);

            mostrarMensajeFinal("EMPATO");
                                    return;
                                }


                                int r, c;
                                do {
                                    r = rand() % 3;
                                    c = rand() % 3;
                                } while (tablero[r][c] !=
' ');
                                tablero[r][c] = cpu;
                                jugadas++;


                                if
(verificarGanador(tablero, cpu)) {
                                    system("cls");

            dibujarMarco();

            dibujarTablero(tablero);
            resaltarJugadaGanadora(tablero, cpu);  //

            mostrarMensajeFinal("PERDIO");
                                    return;
                                }


                                if (jugadas == 9) {
                                    system("cls");

            dibujarMarco();

            dibujarTablero(tablero);
                                    Sleep(5000);

            mostrarMensajeFinal("EMPATO");
                                    return;
                                }
                            }
                        }
                    }
                }

void jugarMedio() {
            pantallaCarga();
            char tablero[3][3] = { {' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', '
'} };
            int fila = 0, col = 0, jugadas = 0;
            char jugador, cpu;
            jugador = rand() % 2 == 0 ? 'X' : 'O';
            cpu = jugador == 'X' ? 'O' : 'X';
```

```
    if (rand() % 2 == 1) {
        int r, c;
        do {
            r = rand() % 3;
            c = rand() % 3;
        } while (tablero[r][c] != ' ');
        tablero[r][c] = cpu;
        jugadas++;
    }

    while (1) {
        system("cls");
        dibujarMarco();
        dibujarTablero(tablero);
        int x = 5 + col * 11 + 4;
        int y = 6 + fila * 6 + 2;
        gotoxy(x, y);

        char tecla = toupper(getch());
        if (tecla == 'W' && fila > 0) fila--;
        else if (tecla == 'S' && fila < 2) fila++;
        else if (tecla == 'A' && col > 0) col--;
        else if (tecla == 'D' && col < 2) col++;
        else if (tecla == '\r') {
            if (tablero[fila][col] == ' ') {
                tablero[fila][col] = jugador;
                jugadas++;

                if (verificarGanador(tablero, jugador)) {
                    system("cls");
                    dibujarMarco();
                    dibujarTablero(tablero);
                    resaltarJugadaGanadora(tablero, jugador);
                    mostrarMensajeFinal("GANO");
                    return;
                }

                if (jugadas == 9) {
                    system("cls");
                    dibujarMarco();
                    dibujarTablero(tablero);
                    Sleep(5000);
                    mostrarMensajeFinal("EMPATO");
                    return;
                }

                cpuMovimientoMedio(tablero, jugador, cpu);
                jugadas++;

                if (verificarGanador(tablero, cpu)) {
                    system("cls");
                    dibujarMarco();
```

```
        dibujarTablero(tablero);

        resaltarJugadaGanadora(tablero, cpu);

        mostrarMensajeFinal("PERDIO");
                                        return;
                                }


                                if (jugadas == 9) {
                                        system("cls");

        dibujarMarco();

        dibujarTablero(tablero);
                                        Sleep(5000);

        mostrarMensajeFinal("EMPATO");
                                        return;
                                }
                        }
                }
        }
}


void jugarDificil() {
        pantallaCarga();
        char tablero[3][3] = { {' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', '
'} };
        int fila = 0, col = 0, jugadas = 0;
        char jugador, cpu;
        jugador = rand() % 2 == 0 ? 'X' : 'O';
        cpu = jugador == 'X' ? 'O' : 'X';

        if (rand() % 2 == 1) {
                int r, c;
                do {
                        r = rand() % 3;
                        c = rand() % 3;
                } while (tablero[r][c] != ' ');


                cpuMovimientoDificil(tablero,     jugador,
cpu);
                jugadas++;
        }


        while (1) {
                system("cls");
                dibujarMarco();
                dibujarTablero(tablero);
                int x = 5 + col * 11 + 4;
                int y = 6 + fila * 6 + 2;
                gotoxy(x, y);


                char tecla = toupper(getch());
                if (tecla == 'W' && fila > 0) fila--;
                else if (tecla == 'S' && fila < 2) fila++;
                else if (tecla == 'A' && col > 0) col--;
                else if (tecla == 'D' && col < 2) col++;
                else if (tecla == '\r') {
                        if (tablero[fila][col] == ' ') {
                                tablero[fila][col]          =
jugador;

                                jugadas++;
```

```
                    if (verificarGanador(tablero, jugador)) {
                        system("cls");
    dibujarMarco();
    dibujarTablero(tablero);
    resaltarJugadaGanadora(tablero, jugador);
    mostrarMensajeFinal("GANO (ERROR)");
                        return;
                    }

                    if (jugadas == 9) {
                        system("cls");
    dibujarMarco();
    dibujarTablero(tablero);
                        Sleep(5000);
    mostrarMensajeFinal("EMPATO");
                        return;
                    }

    cpuMovimientoDificil(tablero, jugador, cpu);
                    jugadas++;

                    if (verificarGanador(tablero, cpu)) {
                        system("cls");
    dibujarMarco();
    dibujarTablero(tablero);
    resaltarJugadaGanadora(tablero, cpu);
    mostrarMensajeFinal("PERDIO");
                        return;
                    }

                    if (jugadas == 9) {
                        system("cls");
    dibujarMarco();
    dibujarTablero(tablero);
                        Sleep(5000);
    mostrarMensajeFinal("EMPATO");
                        return;
                    }
                }
            }
    }
}
```

15) Función para mostrar la pantalla final con nombre de integrantes del grupo:

```
const char* linea1 = "Sergio Alejandro Sagastume Gonzalez Programacion I 202508005";
    const char* linea2 = "Annelis Juany Sacalxot Chojolan Programacion I 202508040";
    int y = 0;
    int alternadorColor = 0;
```

```
    while (y < 26) {
        if (alternadorColor % 2 == 0)
            system("color 5F");
        else
            system("color 2F");

        system("cls");
        gotoxy(10, y);
        printf("%s", linea1);
        gotoxy(10, y + 2);
        printf("%s", linea2);

        Sleep(200);
        y++;
        alternadorColor++;
    }
```

16) Función del menú para elegir dificultad:

```
void menuPrincipal() {
    while (1) {
        system("cls");
        dibujarMarco();
        gotoxy(10, 10); printf("\tJUEGO TOTITO");
        gotoxy(10, 11); printf("1. Jugar Facil");
        gotoxy(10, 12); printf("2. Jugar Medio");
        gotoxy(10, 13); printf("3. Jugar Dificil");
        gotoxy(10, 14); printf("4. Regresar al menu principal");
        gotoxy(10, 15); printf("Selecciona una opcion: ");
        char opcion = toupper(getch());
        switch (opcion) {
        case '1': jugarFacil(); break;
        case '2': jugarMedio(); break;
        case '3': jugarDificil(); break;
        case '4': menuInicio(); // REGRESA a menuInicio
        }
    }
}
```

17) Función menú inicial:

```
void menuInicio() {
    system("cls");
    dibujarMarco();

    for (int i = 0; i < 6; i++) {
        gotoxy(5, 6);
        if (i % 2 == 0) cambiarColor(COLOR_X);
        else cambiarColor(COLOR_O);
        printf("JUEGO DE TOTITO");
        Sleep(300);
    }
    cambiarColor(7);

    gotoxy(5, 10);
    printf("A) ELEGIR DIFICULTAD");
    gotoxy(5, 11);
    printf("B) SALIR");

    char opcion;
    do {
```

```
            gotoxy(5, 13);
            printf("Seleccione una opcion (A/B): ");
            opcion = toupper(getch());
    } while (opcion != 'A' && opcion != 'B');

    if (opcion == 'A') {
            menuPrincipal();
    } else if (opcion == 'B') {
            mostrarSalida();
    }
}
```

18) Función que resalta la jugada ganadora:

```
void resaltarJugadaGanadora(char tablero[3][3], char jugador) {
    int baseX = 5, baseY = 6;
    int i;

    cambiarColor(14);

    for (i = 0; i < 3; i++) {
            if (tablero[i][0] == jugador && tablero[i][1] == jugador && tablero[i][2] == jugador) {
                    for (int j = 0; j < 3; j++) {
                            gotoxy(baseX + j * 11 + 4, baseY + i * 6 + 2);
                            printf("%c", tablero[i][j]);
                    }
                    Sleep(1500);
                    cambiarColor(7);
                    return;
            }
    }

    for (i = 0; i < 3; i++) {
            if (tablero[0][i] == jugador && tablero[1][i] == jugador && tablero[2][i] == jugador) {
                    for (int j = 0; j < 3; j++) {
                            gotoxy(baseX + i * 11 + 4, baseY + j * 6 + 2);
                            printf("%c", tablero[j][i]);
                    }
                    Sleep(1500);
                    cambiarColor(7);
                    return;
            }
    }

    if (tablero[0][0] == jugador && tablero[1][1] == jugador && tablero[2][2] == jugador) {
            for (i = 0; i < 3; i++) {
                    gotoxy(baseX + i * 11 + 4, baseY + i * 6 + 2);
                    printf("%c", tablero[i][i]);
            }
            Sleep(1500);
            cambiarColor(7);
            return;
    }
```

```
    if (tablero[0][2] == jugador && tablero[1][1] == jugador && tablero[2][0] == jugador) {
        for (i = 0; i < 3; i++) {
            gotoxy(baseX + (2 - i) * 11 + 4, baseY + i * 6 + 2);
            printf("%c", tablero[i][2 - i]);
        }
        Sleep(1500);
        cambiarColor(7);
        return;
    }
}
```

19) Función principal del programa:

```
int main() {
    srand(time(NULL));
    menuInicio();
    return 0;
}
```

III) LÓGICA DEL JUEGO:

1) Menú principal: En este apartado se le presenta al jugador donde puede elegir la dificultad.

Figura 1: Menú de inicio

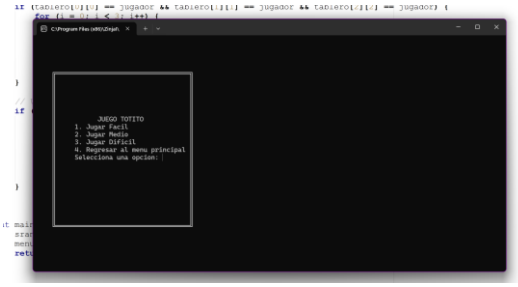


Figura 2: Menú de dificultad

2) Tablero de Juego: Cuadricula de 3x3 donde se visualiza las jugadas del usuario y de la máquina. Se maneja a través de coordenadas para la ubicación de los símbolos ya sea X o O.

Figura 3: Tablero de juego

3) Ciclo de Juego: El juego asigna de manera aleatoria quien inicia primero, si el jugador o la máquina. Posteriormente se alternan hasta que uno de los dos consigue la jugada ganadora o los 9 espacios del tablero son ocupados sin que se consiga a un ganador, por lo que se toma como empate.

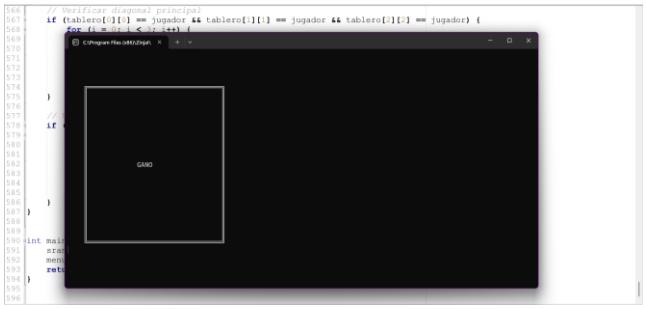


Figura 4. Pantalla de victoria.

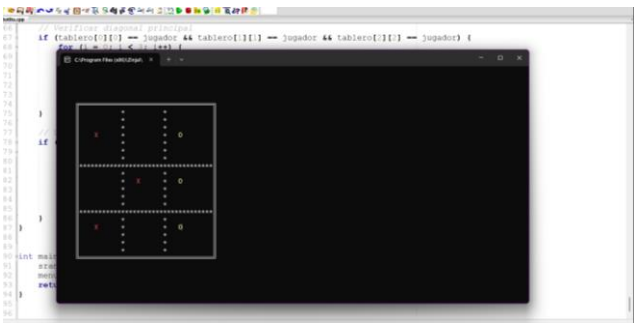
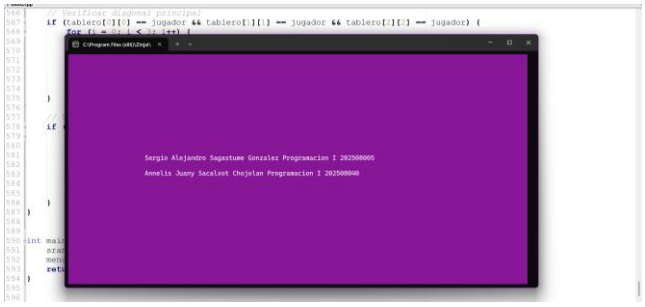


Figura 5. Jugada ganadora resaltada

4) Pantalla final con el nombre de los integrantes del grupo:

Aquí, al pulsar salir en el menú de seleccionar dificultad, se despliega los nombres de los integrantes del equipo.



5) Variables utilizadas para funcionamiento del juego:

- **char tablero[3][3]:** Representa el estado actual del tablero, con 9 espacios donde se colocan 'X', 'O' o ' ' (vacío).
- **int fila = 0, col = 0:** Posición actual del cursor del jugador en el tablero (fila y columna). Permite al jugador moverse por el tablero con las teclas W, A, S, D para seleccionar una casilla.
- **int jugadas = 0:** Lleva la cuenta del número total de jugadas realizadas (por ambos jugadores). Es esencial para detectar empates (cuando jugadas == 9) y controlar el flujo del juego.
- **char jugador, cpu:** Almacenan los símbolos del jugador humano ('X' o 'O') y de la CPU (el símbolo opuesto). Permiten distinguir entre los dos jugadores y aplicar las reglas según el turno.
- **int r, c:** Coordenadas aleatorias para el primer movimiento de la CPU, si le toca iniciar. Se usan para asegurar que la CPU escoja una casilla libre al azar al principio.
- **int x, y:** Coordenadas en pantalla para posicionar el cursor visual en la consola. Usadas con gotoxy(x, y) para mover el cursor al lugar correcto donde el jugador quiere poner su ficha.
- **char tecla:** Captura la tecla presionada por el usuario (convertida a mayúscula). Permite detectar los movimientos (W, A, S, D) y la selección (Enter).

IV. CONCLUSIONES

El desarrollo del juego TOTITO en C reveló cómo la lógica de cada jugada refleja la toma de decisiones en contextos complejos, tanto reales como estratégicos. Al igual que un jugador analiza su movimiento en un tablero limitado, el proyecto aplicó principios de organización, control y anticipación propios de la programación estructurada. Esta forma de programar en consola, aunque elemental en apariencia al usar solo texto, colores y teclas, exige precisión técnica. Cada interacción del usuario ejemplifica un proceso lógico que estimula el razonamiento estratégico, la resolución de problemas y la abstracción, trascendiendo el simple entretenimiento.

Lejos de recurrir a librerías modernas o entornos visuales avanzados, este proyecto hace uso intencional de herramientas básicas del lenguaje C y funciones especificas como gotoxy() y set, recordando que en la sencillez ConsoleCursorPosition(). Al igual que ciertas obras de ciencia ficción utilizan tecnología "retro" para dar lugar a nuevas narrativas, aquí se explora la programación de consola como medio para generar un sistema interactivo que combina diseño, lógica y experiencia del usuario.

En definitiva, este juego representa más que un ejercicio académico: es una oportunidad para entender cómo la programación puede simular desafíos reales, desarrollar pensamiento estructurado, y ofrecer soluciones creativas con recursos limitados.

V. VIDEO DEMOSTRATIVO

Para mostrar cómo funciona el juego TOTITO que se programó en lenguaje C, se preparó un video demostrativo donde se puede ver desde que el juego inicia hasta que termina una partida completa. En el video se enseña cómo moverse por el tablero usando las teclas W, A, S y D, cómo se elige la dificultad del juego y cómo se colocan las fichas con la tecla ENTER.

También se puede ver cómo responde la computadora dependiendo del nivel (fácil, medio o difícil), y cómo el programa detecta cuándo alguien gana o si hay un empate.

Además, se muestra el uso de colores para diferenciar las jugadas del jugador y de la CPU, y cómo se resalta la jugada ganadora cuando alguien logra hacer tres en línea.

El enlace del video es: Video demostrativo totito