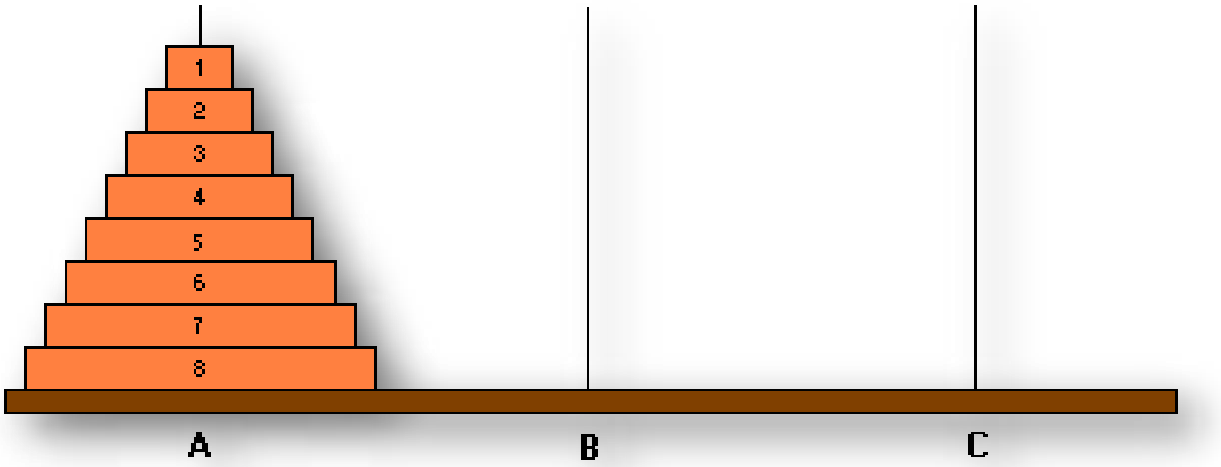
TORRES DE HANOI

Proyecto Final



Anguiano Reséndez Leslie Andrea, 70930

José Francisco Estañón Miranda, 70243

Facultad de Negocios

Actuaría 301

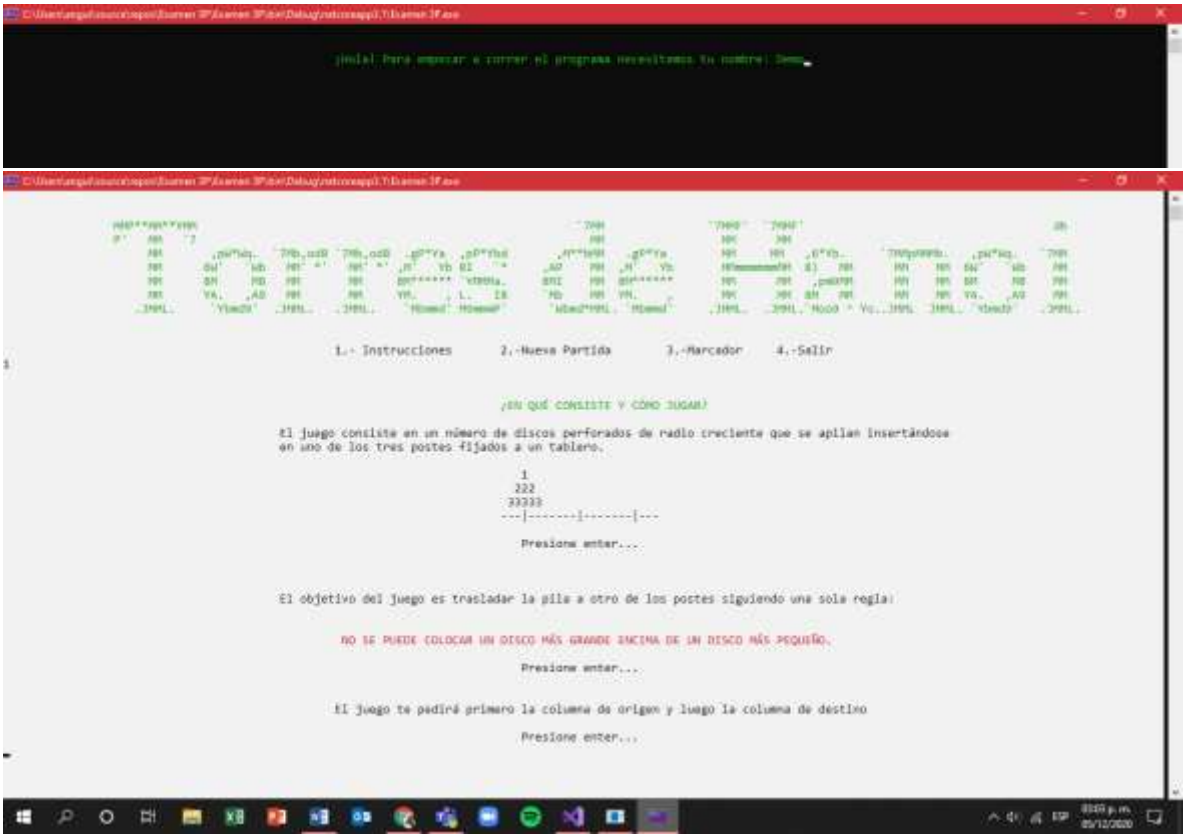
INTRODUCCIÓN

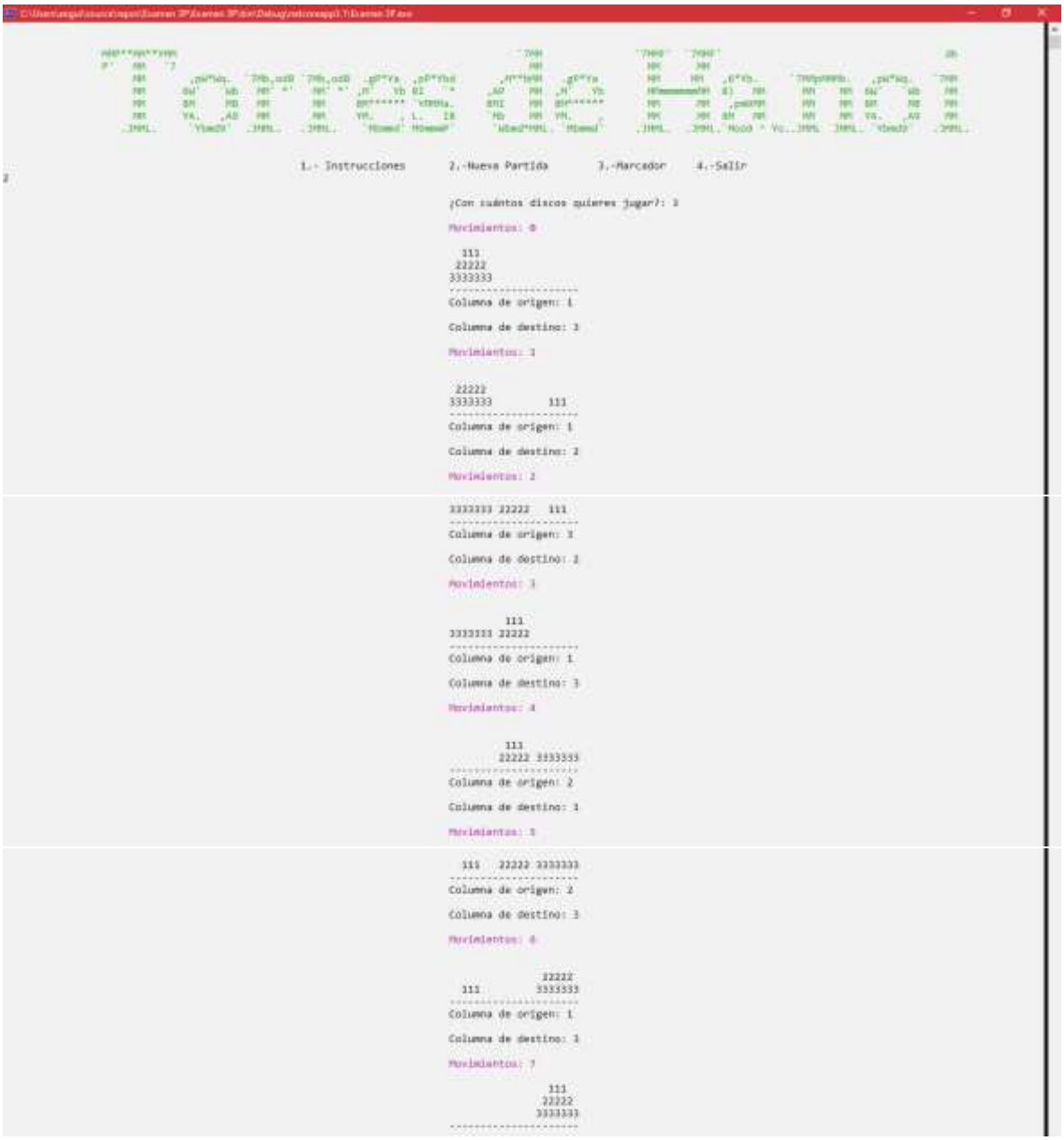
Para nuestro proyecto final decimos hacer el código para el juego de las **Torres de Hanói**.

Las Torres de Hanói es un rompecabezas o juego matemático inventado en 1883 por el matemático francés Édouard Lucas. Este juego de consiste en un número de discos perforados de radio creciente que se apilan insertándose en uno de los tres postes fijados a un tablero. El objetivo del juego es trasladar la pila a otro de los postes siguiendo ciertas reglas, como que no se puede colocar un disco más grande encima de un disco más pequeño.

A continuación, se muestra al código funcionando y el pseudocódigo.

DESARROLLO DE LA PRÁCTICA




C:\Users\angel\source\repos\Examen 3P\Examen 3P\bin\Debug\net6.0\examen3p.exe
1.- Instrucciones
2.-Nueva Partida
3.-Marcador
4.-Salir
2
¿Con cuántos discos quieres jugar?: 3
Movimientos: 0
111
22222
3333333
Columna de origen: 1
Columna de destino: 3
Movimientos: 1
22222
3333333 111
Columna de origen: 1
Columna de destino: 2
Movimientos: 2
3333333 22222 111
Columna de origen: 3
Columna de destino: 2
Movimientos: 3
111
3333333 22222
Columna de origen: 1
Columna de destino: 3
Movimientos: 4
111
22222 3333333
Columna de origen: 2
Columna de destino: 1
Movimientos: 5
111 22222 3333333
Columna de origen: 2
Columna de destino: 3
Movimientos: 6
22222
111 3333333
Columna de origen: 1
Columna de destino: 3
Movimientos: 7
111
22222
3333333


Movimientos: 7
¿Quieres guardar tu puntuación?
Se ha guardado exitosamente.
Presione enter...


Nuevo Movimientos: 7 Discos: 3 Fecha: 05/12/2020 12:00:00 a. m.
Guardado Movimientos: 7 Discos: 3 Fecha: 05/12/2020 12:00:00 a. m.
Demo Movimientos: 7 Discos: 3 Fecha: 05/12/2020 12:00:00 a. m.
Presione enter...


1.- Instrucciones 2.-Nueva Partida 3.-Marcador 4.-Salir
4
C:\Users\angul\source\repos\Examen 3P\Examen 3P\bin\Debug\netcoreapp3.1\Examen 3P.exe (proceso 14892) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->Cerrar la consola automáticamente al detenerse l
a depuración.
Presione cualquier tecla para cerrar esta ventana. . .

PSEUDOCÓDIGO:

```
/*Jose Francisco Estañon Miranda 70243
  Leslie Andrea Anguiano Reséndez 70930*/
using System;
using System.Linq;
using System.IO;

namespace Torres_de_Hanoi
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.ForegroundColor = ConsoleColor.Green;
            Console.Write("\n\n\t\t\t\t\t\t¡Hola! Para empezar a correr el programa
necesitamos tu nombre: ");
            string usuario = Console.ReadLine();

            Menu(usuario);
        }

        public static void Menu(string usuario)
        {
            if (File.Exists("marcador.txt") == false)
            {
                using (StreamWriter sw = File.AppendText("marcador.txt"))
                {
                    sw.Close();
                }
            }

            Console.BackgroundColor = ConsoleColor.White;
            Console.ForegroundColor = ConsoleColor.DarkGreen;

            int opc = 2;
            Console.Clear();

            Console.WriteLine("\n\n\t\tMMP**MM**YMM
`7MM               `7MMF'  `7MMF'                                     db  ");
            Console.WriteLine("\t\tP'   MM   `7
MM                 MM      MM                                        ");
            Console.WriteLine("\t\t     MM       ,pW*Wq.  `7Mb,od8 `7Mb,od8  .gP*Ya
,pP*Ybd       ,M**bMM   .gP*Ya        MM     MM   ,6*Yb.     `7MMpMMMb.   ,pW*Wq.   `7MM
");
            Console.WriteLine("\t\t     MM      6W'   `Wb   MM' *'   MM' *' ,M'   Yb 8I
`*     ,AP    MM  ,M'   Yb       MMmmmmmmMM  8)   MM      MM    MM  6W'   `Wb    MM  ");
            Console.WriteLine("\t\t     MM      8M     M8   MM       MM     8M******
`YMMMa.     8MI    MM  8M******       MM      MM   ,pm9MM      MM    MM  8M     M8    MM
");
            Console.WriteLine("\t\t     MM      YA.   ,A9   MM       MM     YM.    , L.
I8     `Mb    MM  YM.    ,       MM      MM  8M   MM      MM    MM  YA.   ,A9    MM  ");
            Console.WriteLine("\t\t   .JMML.     `Ybmd9'  .JMML.   .JMML.    `Mbmmd'
M9mmmP'      `Wbmd*MML. `Mbmmd'     .JMML.  .JMML.`Moo9 ^ Yo..JMML  JMML. `Ybmd9'
.JMML.");

            Console.ForegroundColor = ConsoleColor.Black;
            Console.WriteLine("\n\n\t\t\t\t\t\t1.- Instrucciones\t2.-Nueva Partida\t3.-
Marcador\t4.-Salir");
```

```
            try
            {
                opc = int.Parse(Console.ReadLine());
            }
            catch (Exception) { Menu(usuario); }
            switch (opc)
            {
                case 1:
                    try
                    {
                        Instrucciones(usuario);
                    }
                    catch (Exception) { Menu(usuario); }
                    break;

                case 2:
                    try
                    {
                        Console.Write("\n\t\t\t\t\t\t\t\t\t¿Con cuántos discos quieres
jugar?: ");
                        int discos = int.Parse(Console.ReadLine());

                        if (discos == 0)
                        {
                            Console.ForegroundColor = ConsoleColor.DarkRed;
                            Console.WriteLine("\t\t\t\t\t\t\t\t\tReally?     -_- ");
                            Console.ForegroundColor = ConsoleColor.Black;
                        }

                        while (discos > 20)
                        {
                            Console.ForegroundColor = ConsoleColor.DarkRed;
                            Console.WriteLine("\t\t\t\t\t\t\t\t\tPor favor se razonable,
esto dificultaría visualizarlo y ejecutarlo");
                            Console.ForegroundColor = ConsoleColor.Black;
                            Console.WriteLine("\t\t\t\t\t\t\t\t\tInténtalo de nuevo: ");
                            discos = int.Parse(Console.ReadLine());
                        }

                        int movimientos = 0;
                        int[,] Partida = Nuevo(discos);
                        Dibujar(Partida, discos, movimientos);
                        bool ganar;

                        do
                        {
                            movimientos++;
                            Partida = Juego(Partida, discos, movimientos);
                            ganar = Ganar(Partida, discos);
                        } while (ganar == false);
                        Felicidades(movimientos, discos, usuario);

                    }
                    catch (Exception)
                    {
                        Console.ForegroundColor = ConsoleColor.DarkRed;
                        Console.WriteLine("\t\t\t\t\t\t\t\t\tVaya algo salió mal... De
vuelta al Menú");
                        Console.ForegroundColor = ConsoleColor.Black;
                        Console.WriteLine("\t\t\t\t\t\t\t\t\tPresiona enter...");
                        Console.ReadLine();
```

```
                        Menu(usuario);
                    }
                    break;

                case 3:
                    try
                    {
                        Marcador(usuario);
                    }
                    catch (Exception) { Menu(usuario); }
                    break;

                case 4:
                    Environment.Exit(0);
                    break;
                default:
                    Menu(usuario);
                    break;
            }

        }
        public static void Instrucciones(string usuario)
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine("\n\n\t\t\t\t\t\t\t\t\t¿EN QUÉ CONSISTE Y CÓMO JUGAR?");
            Console.ForegroundColor = ConsoleColor.Black;
            Console.WriteLine("\n\t\t\t\t\tEl juego consiste en un número de discos
perforados de radio creciente que se apilan insertándose  \n\t\t\t\ten uno de los tres
postes fijados a un tablero. ");
            Console.WriteLine("\n\t\t\t\t\t\t\t\t\t   1  ");
            Console.WriteLine("\t\t\t\t\t\t\t\t  222 ");
            Console.WriteLine("\t\t\t\t\t\t\t\t 33333");
            Console.WriteLine("\t\t\t\t\t\t\t\t---|-------|-------|---");
            Console.WriteLine("\n\t\t\t\t\t\t\t\t   Presione enter...");
            Console.ReadLine();

            Console.WriteLine("\n\n\t\t\t\t\tEl objetivo del juego es trasladar la pila a
otro de los postes siguiendo una sola regla: ");
            Console.ForegroundColor = ConsoleColor.DarkRed;
            Console.WriteLine("\n\n\t\t\t\t\t NO SE PUEDE COLOCAR UN DISCO MÁS GRANDE
ENCIMA DE UN DISCO MÁS PEQUEÑO.");
            Console.ForegroundColor = ConsoleColor.Black;
            Console.WriteLine("\n\t\t\t\t\t\t\t\t   Presione enter...");
            Console.ReadLine();

            Console.WriteLine("\n\t\t\t\t\t\tEl juego te pedirá primero la columna de
origen y luego la columna de destino");
            Console.WriteLine("\n\t\t\t\t\t\t\t\t   Presione enter...");
            Console.ReadLine();

            Console.WriteLine("\n\t\t\t\t\t\tEso sería todo, ¿listo para jugar?");
            Console.WriteLine("\n\t\t\t\t\t\t\t\t   Presione enter...");
            Console.ReadLine();
            Menu(usuario);
        }
        public static void Dibujar(int[,] H, int discos, int movimientos)
        {
            Console.ForegroundColor = ConsoleColor.Magenta;
            Console.WriteLine("\n\t\t\t\t\t\t\t\tMovimientos: " + movimientos + "\n");
            Console.ForegroundColor = ConsoleColor.Black;
            for (int j = 0; j < discos; j++)
```

```
            {
                Console.Write("\t\t\t\t\t\t\t\t\t");
                for (int i = 0; i < 3; i++)
                {
                    if (H[j, i] != 0)
                    {
                        for (int k = 0; k < (discos - H[j, i]); k++) { Console.Write("
"); } // Separacion izquierda

                        string pieza = string.Concat(Enumerable.Repeat(H[j, i], 2 * (H[j,
i] + 1) - 1)); // Numeros
                        Console.Write(pieza);

                        int faltante = (2 * discos) - (((discos - H[j, i]) + (2 * (H[j,
i] + 1) - 1))); // Completa la forma
                        for (int k = 0; k < faltante + 1; k++) { Console.Write(" "); }
                    }

                    else
                    {
                        for (int k = 0; k < 2 * discos + 1; k++) { Console.Write(" "); }
// En caso de que no haya nada
                    }
                }
                Console.Write("\n");
            }
            Console.Write("\t\t\t\t\t\t\t\t\t");
            Console.Write(string.Concat(Enumerable.Repeat("-", 2 * (3 * discos + 1) +
1)));
        }
        public static int[,] Nuevo(int discos)
        {
            int[,] H = new int[discos, 3];

            for (int j = 0; j < 3; j++)
            {
                for (int i = 0; i < discos; i++)
                {
                    if (j == 0)
                    {
                        H[i, j] = i + 1;
                    }
                    else { H[i, j] = 0; }
                }
            }
            return (H);
        }
        public static int[,] Juego(int[,] H, int discos, int movimientos)
        {
            Console.Write("\n\t\t\t\t\t\t\t\t\tColumna de origen: ");
            int C_orig = int.Parse(Console.ReadLine()) - 1;
            Console.Write("\n\t\t\t\t\t\t\t\t\tColumna de destino: ");
            int C_dest = int.Parse(Console.ReadLine()) - 1;


            // *** Logica del juego ***
            if (C_orig != C_dest)
            {
                for (int i = 0; i < discos; i++)
// Itera en todas las filas
                {
```

```
                    if (H[i, C_orig] != 0)
// Busca el Primer disco
                    {
                        for (int j = discos - 1; j >= 0; j--)
// Itera en la columna destino de abajo hacia arriba
                        {
                            if (H[j, C_dest] == 0)
// Busca un espacio libre
                            {
                                if (j < discos - 1)
// Si no es el unico disco
                                {
                                    if (H[i, C_orig] > H[j + 1, C_dest])
// Compara con el de abajo
                                    {
                                        Console.ForegroundColor = ConsoleColor.DarkRed;
                                        Console.WriteLine("\t\t\t\t\t\t\t\t\tEse es un
movimiento ilegal");   // Muestra si se puede *Evitar que cuente
                                        Console.ForegroundColor = ConsoleColor.Black;
                                        break;
                                    }
                                    else
// Realiza el cambio
                                    {
                                        H[j, C_dest] = H[i, C_orig];
                                        H[i, C_orig] = 0;
                                        break;
                                    }
                                }
                                else
                                {
                                    H[j, C_dest] = H[i, C_orig];
                                    H[i, C_orig] = 0;
                                    break;
                                }
                            }
                        }
                        break;
                    }
                    else if (i == (discos - 1)) // Si no encuentra nada sale *Evitar que
cuente*
                    {
                        Console.ForegroundColor = ConsoleColor.DarkRed;
                        Console.WriteLine("\t\t\t\t\t\t\t\t\t¡En esa columna no hay nada!
");
                        Console.ForegroundColor = ConsoleColor.Black;
                        break;
                    }
                }
            }
            else {
                Console.ForegroundColor = ConsoleColor.DarkRed;
                Console.WriteLine("\t\t\t\t\t\t\t\t\t No tiene mucho sentido mover a la
misma columna...");
                Console.ForegroundColor = ConsoleColor.Black;
            }
            // ** termina logica del juego **

            Dibujar(H, discos, movimientos);
            return (H);
        }
```

```
        public static Boolean Ganar(int[,] H, int discos)
        {
            for (int i = 0; i < discos; i++)
            {
                if (H[i, 2] != (i + 1))
                {
                    return false;
                }
            }
            return true;
        }
        public static void Felicidades(int movimientos, int discos, string usuario)
        {
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            int optimo = (int)Math.Pow(2, discos) - 1;

            Console.WriteLine("\n\n\t\t\t\t\t`7MM***YMM          `7MM    db
db       `7MM                `7MM                    OO   ");
            Console.WriteLine("\t\t\t\t\t  MM    `7            MM
MM                  MM                    88   ");
            Console.WriteLine("\t\t\t\t\t  MM   d    .gP*Ya    MM  `7MM   ,p6*bo  `7MM
,M**bMM   ,6*Yb.    ,M**bMM   .gP*Ya   ,pP*Ybd  ||  ");
            Console.WriteLine("\t\t\t\t\t  MM**MM   ,M'   Yb   MM    MM  6M'  OO    MM,
AP     MM  8)   MM  ,AP    MM, M'    Yb  8I   `*  ||  ");
            Console.WriteLine("\t\t\t\t\t  MM   Y   8M******   MM    MM  8M         MM
8MI    MM   ,pm9MM  8MI    MM  8M******  `YMMMa.  `'  ");
            Console.WriteLine("\t\t\t\t\t  MM       YM.    ,   MM    MM  YM.    ,   MM
`Mb    MM  8M   MM  `Mb    MM  YM.    ,  L.   I8  ,,  ");
            Console.WriteLine("\t\t\t\t\t.JMML.      `Mbmmd' .JMML..JMML. YMbmd'  .JMML.
`Wbmd*MML.`Moo9^Yo. `Wbmd*MML. `Mbmmd'  M9mmmP'  db  ");

            Console.WriteLine("\n\t\t\t\t\t\t\t\t\t\t" + usuario + " hiciste: " +
movimientos + " movimientos");
            Console.WriteLine("\t\t\t\t\t\t\t\t\tEl mínimo número de movimietos es: " +
optimo);
            Console.WriteLine("\t\t\t\t\t\t\t\t\tPresione enter...");
            Console.ReadLine();

            if (movimientos == optimo)
            {

                Console.WriteLine("\n\n\t\t\t\t\t\t\t\t\t\t          ccccccccccccc
");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t          c cccccccccc c      ");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t        c    ccccccccc    c    ");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t           c  ccccccc  c      ");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t              cccc            ");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t               cc             ");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t          ccccccccccccc       ");
                Console.WriteLine("\t\t\t\t\t\t\t\t\t          ccccccccccccc       ");

                Console.WriteLine("\n\t\t\t\t\t\t\t\t\t\t!Felicidades " + usuario + ",
lograste un buen puntuaje!");
                Console.ReadLine();

                int guardar;
                Console.WriteLine("\t\t\t\t\t\t\t\t\t¿Quieres guardar tu
puntuación?\n\t\t\t\t\t\t\t\t\t1.-Sí\t2.-No");
                guardar = int.Parse(Console.ReadLine());
                if (guardar == 1)
                {
```

```
                    string linea = usuario + "\tMovimientos: " + movimientos + "\tDiscos:
" + discos + "\tFecha: " + DateTime.Today;
                    using (StreamWriter sw = File.AppendText("marcador.txt"))
                    {
                        sw.WriteLine(linea);
                        sw.Close();
                    }

                    Console.WriteLine("\t\t\t\t\t\t\t\t\tSe ha guardado exitosamente.");
                    Console.WriteLine("\n\n\t\t\t\t\t\t\t\t\tPresione enter...");
                    Console.ReadLine();
                }
                else
                {
                    Console.WriteLine("\n\n\t\t\t\t\t\t\t\t\tPresione enter...");
                    Console.ReadLine();
                }
            }
            Menu(usuario);
        }
        public static void Marcador(string ususario)
        {
            Console.Clear();
            Console.ForegroundColor = ConsoleColor.DarkRed;

            Console.WriteLine("\n\n\t\t\t\t\t
,,                              ccccccccccccccc");
            Console.WriteLine("\t\t\t\t\t  7MMM.    ,MMF
`7MM                             c  ccccccccccc  c");
            Console.WriteLine("\t\t\t\t\t  MMMb     PMM
MM                              c   ccccccccc   c");
            Console.WriteLine("\t\t\t\t\t  M YM,   M MM   ,6'Yb.  `7Mb,od8  ,p6'bo
,6'Yb.    ,MMCbMM   ,pW'Wq.  `7Mb,od8           c  ccccccc  c");
            Console.WriteLine("\t\t\t\t\t  M  Mb  M' MM  8)   MM    MM'    6M   OO  8)
MM  ,AP    MM  6W'    Wb   MM   '               cccc");
            Console.WriteLine("\t\t\t\t\t  M  YM.P'  MM   ,pm9MM    MM     8M
,pm9MM  8MI    MM  8M     M8   MM                    cc");
            Console.WriteLine("\t\t\t\t\t  M  `YM'   MM  8M   MM    MM     YM.    , 8M
MM  `Mb    MM  YA.   ,A9  MM               ccccccccccccc");
            Console.WriteLine("\t\t\t\t\t.JML. `'  .JMML.`Moo9^Yo..JMML.    YMbmd'  `Moo9
^ Yo. `Wbmd' MM'Ybmd9'.JMML.              ccccccccccccc");

            string linea;
            using (StreamReader sr = new StreamReader("marcador.txt"))
            {
                linea = sr.ReadToEnd();
                Console.WriteLine(linea);
                sr.Close();
            }

            Console.WriteLine("\n\n\t\t\t\t\t\t\t\t\tPresione enter...");
            Console.ReadLine();
            Menu(ususario);

        }

    }
}
```

CONCLUSIONES:

En el proyecto tuvimos la oportunidad de reforzar lo aprendido y ponerlo en práctica de una manera divertida.

Utilizamos cambios de color y texto en ASCII para darle una interfaz más amigable para el usuario. Las funciones fueron de mucha ayuda con este tipo de código, ya que te ahorras demasiadas líneas de código, evitando repetir cada uno de los procedimientos. Los métodos burbuja también fueron pieza clave del código para poder hacer el juego.

Está materia nos ha ayudado a conocer la programación de cerca y crear herramientas útiles para nuestro estudio y vida profesional.