| Logo instituto sin fondo |
| --- |
| **INSTITUTO SUPERIOR DE FORMACIÓN TÉCNICA N° 93****CARRERA:** Tecnicatura superior en análisis, desarrollo y programación de aplicaciones**ESPACIO CURRICULAR:** Programacion I**CURSO:** 1 AÑO B**CICLO LECTIVO:** 2024**PROFESOR:** Pini LeandroPLAN AUTORIZADO POR RESOLUCIÓN N°6175/03 |
| **TP INTEGRADOR ANUAL****ALUMNO:** Nicolás Sandoval**CANTIDAD DE HOJAS:** **FECHA DE ENTREGA:** 2024/10/30 |

#### 

#### 

#### 

#### 

#### 

**1-** Desarrollar un algoritmo que cargue una matriz de 4 x 4 con valores enteros aleatorios (Random) comprendidos

entre 1 y 6. Luego de cargadas la misma, el programa debe hacer lo siguiente:

a- Imprimir la matriz en consola.

b- Recorrerla posición a posición, ir sumando el valor de cada celda y luego de terminar de recorrerla

imprimir en consola, y debajo de la matriz ya impresa, el resultado de esa suma de todos los valores.

| **Ejercicio 1** | |
| --- | --- |
| using System;  namespace ejercicio1  {  public static class Program  {  private static void Main(string[] args)  {  int[,] matriz = new int[4, 4];  int suma = 0;  Random random = new Random();  for (var i = 0; i < 4; i++)  {  for (var j = 0; j < 4; j++)  {  matriz[i, j] = random.Next(1, 7);  Console.Write(matriz[i, j]+" ");  suma += matriz[i, j];  }  Console.WriteLine("");  }  Console.WriteLine("La suma de los valores de la matriz es: "+suma);  Console.ReadKey();  }  }  } |  |

**2-** Generar un algoritmo que cargue una matriz de 5 x 5 (con el formato y valores de la imagen que se muestra a

continuación mediante un algoritmo automático) y que al final de cada fila muestre la sumatoria de los valores

que forman las mismas:

| **Ejercicio 2** | |
| --- | --- |
| using System;  namespace ejercicio2  {  static class Program  {  private static void Main(string[] args)  {  int[,] matriz = new int[5, 5];  int suma = 0;  for (var i = 0; i < 5; i++)  {  for (var j = 0; j < 5; j++)  {  matriz[i, j] = i + j;  suma += matriz[i, j];  Console.Write(matriz[i,j] + ” ");  }  Console.WriteLine("= “ + suma);  suma = 0;  }  }  }  } |  |

**3-** Elaborar un programa que lea o recorra una matriz, que ya debe estar cargadas con valores, y que imprima por

pantalla poniendo las filas como columnas y las columnas como filas (lo que se conoce como matriz transpuesta).

| **Ejercicio 3** | |
| --- | --- |
| using System;  namespace ejercicio3  {  static class Program  {  private static void Main(string[] args)  {  int[,] ints = { { 0, 0, 2, 5, 4 }, { 5, 0, 6, 2, 2 }, { 5, 2, 3, 6, 4 } };    for (var i = 0; i < 3; i++)  {  for (var j = 0; j < 5; j++)  {  Console.Write(ints[i,j]+” ");  }  Console.WriteLine("");  }  Console.WriteLine("");  for (var i = 0; i < 5; i++)  {  for (var j = 0; j < 3; j++)  {  Console.Write(ints[j,i]+” ");  }  Console.WriteLine("");  }  }  }  } |  |

**4-** Elabore un programa que cargue una matriz de 6 x 6 con números aleatorios entre 2 y 8. Imprima la misma por

pantalla. Al mismo tiempo que la imprime debe almacenar en un array de 6 posiciones los valores de la diagonal

de dicha matriz (diagonal de izquierda a derecha y de arriba hacia abajo) y luego se debe imprimir dicho array.

| **Ejercicio 4** | |
| --- | --- |
| using System;  namespace ejercicio4  {  static class Program  {  private static void Main(string[] args)  {  int[,] ints = new int[6, 6];  int[] ints1 = new int[6];  Random random = new Random();  for (var i = 0; i < 6; i++)  {  for (var j = 0; j < 6; j++)  {  ints[i, j] = random.Next(2, 9);  Console.Write(ints[i, j] + ” ");  if (i == j)  ints1[i] = ints[i, j];  }  Console.WriteLine("");  }  Console.WriteLine("");  foreach (int Item in ints1)  {  Console.Write(Item + ” ");  }  Console.WriteLine("");  }  }  } |  |

**5** - Desarrolle un programa que:

a. Utilice dos arrays unidimensionales de 5 posiciones.

b. Los arrays deberán ser cargados con valores random de entre 1 y 9.

c. Luego deberá tomar cada valor del array1 e ir multiplicando por cada uno de los del array2

| **Ejercicio 5** | |
| --- | --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TPAnualEjer5  {  static class Program  {  static void Main(string[] args)  {  Funs.App();  }  }  }  //////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TPAnualEjer5  {  public static class MisVars  {  public static int[] nums = new int[5];  public static int[] nums2 = new int[5];  public static Random NRandom = new Random();  public static string Archivo = "historial.txt";  public static StringBuilder Sb = new StringBuilder();  }  }  //////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.IO;  namespace TPAnualEjer5  {  public static class AccesoArchivo  {  public static StreamWriter Sw = File.AppendText(MisVars.Archivo);  }  }  //////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TPAnualEjer5  {  public static class FunAccesoArchivo  {  public static void EscribirTxt(StringBuilder Sb)  {  AccesoArchivo.Sw.WriteLine("\n"+ DateTime.Now + "\n" + Sb);  AccesoArchivo.Sw.Flush();  }  public static void CerrarTxt()  {  AccesoArchivo.Sw.Close();  }  }  }  //////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TPAnualEjer5  {  public static class Funs  {  public static void App()  {  CargarArrayRandom("Array 1: ", MisVars.nums);  CargarArrayRandom("Array 2: ", MisVars.nums2);  MultiplicarArray();  FunAccesoArchivo.EscribirTxt(MisVars.Sb);  FunAccesoArchivo.CerrarTxt();  Console.ReadKey(true);  }  public static void CargarArrayRandom(string nombre, int[] array)  {  MisVars.Sb.Append(nombre);  Console.Write(nombre );  for (int i = 0; i < 5; i++)  {  array[i] = MisVars.NRandom.Next(1, 10);  MisVars.Sb.Append(array[i] + " ");  Console.Write(array[i] + " ");  }  MisVars.Sb.Append("\n");  Console.WriteLine("\n");  }    public static void MultiplicarArray()  {  MisVars.Sb.Append("Resultante: ");  Console.Write("Resultante: ");  for (int i = 0; i < 5; i++)  {  for (int j = 0; j < 5; j++)  {  MisVars.nums[i] \*= MisVars.nums2[j];  }  MisVars.Sb.Append(MisVars.nums[i] + " ");  Console.Write(MisVars.nums[i] + " ");  }  }  }  } |  |
| **Ejercicio 6:** Desarrollar un programa que utilice un array bidimensional que permita cargar Nombre, Apellido y Edad de 5 personas (y en ese orden). Luego de que el mismo es cargado con la ayuda o interacción del usuario, el programa deberá recorrer el array e imprimir por consola el contenido de este respetando el formato que se muestra en la siguiente imagen ejemplo de ejecución: | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Text.RegularExpressions;  namespace TPAnualEjer6  {  public static class Funs  {  public static void OpcionInvalida(string frase)  {  Console.Clear();  Colores.ConsolaLetraColorRed();  Console.WriteLine(frase + "\n");  }  public static void Titulo(string fraseIn)  {  Colores.ConsolaLetraColorMagenta();  Console.WriteLine(fraseIn);  }  public static void MenuPrincipal()  {  Colores.ConsolaLetraColorGreen();  Console.WriteLine("\nIngrese una opcion segun su indice y oprima enter\n");  Colores.ConsolaLetraColorWhite();  Console.WriteLine("1- Ingresar personas");  Console.WriteLine("2- Ver Personas");  Console.WriteLine("0- Salir");  }  public static void App()  {  string opcion = "";  do  {  MenuPrincipal();  opcion = Console.ReadLine() ?? " ";  SeleccionOpcion(opcion);  } while (opcion != "0");  }  public static void SeleccionOpcion(string opcion)  {  Console.Clear();  switch (opcion)  {  case "1":  IngresarPersonas();  break;  case "2":  VerPersonas();  break;  case "0": return;  default:  OpcionInvalida("Opcion Invalida");  break;  }  }  public static void VerPersonas()  {  Titulo("PERSONAS\n");  Colores.ConsolaLetraColorGray();  Ordenamiento();  for (int i = 0; i < 5; i++)  Console.WriteLine(i + 1 + "-" + MisVars.personas[i, 1] + ", " + MisVars.personas[i, 0] + " " + MisVars.personas[i, 2]);  }  public static void Ordenamiento()  {  for (int i = 0; i < 5; i++)  {  for (int j = 0; j < 5 - i - 1; j++)  {  int per1 = int.Parse(MisVars.personas[j, 2]);  int per2 = int.Parse(MisVars.personas[(j + 1), 2]);  if (per1 > per2)  {  string temp2 = MisVars.personas[j + 1, 2];  string temp1 = MisVars.personas[j + 1, 1];  string temp0 = MisVars.personas[j + 1, 0];  MisVars.personas[j + 1, 2] = MisVars.personas[j, 2];  MisVars.personas[j + 1, 1] = MisVars.personas[j, 1];  MisVars.personas[j + 1, 0] = MisVars.personas[j, 0];  MisVars.personas[j, 2] = temp2;  MisVars.personas[j, 1] = temp1;  MisVars.personas[j, 0] = temp0;  }  }  }  }  public static void IngresarPersonas()  {  for (int f = 0; f < 5; f++)  {  Console.Clear();  Titulo("INGRESO DE PERSONAS\n");    MisVars.temp[0] = IngresoPalabra("Nombre", (f + 1));  if (MisVars.temp[0] == " ") return;  MisVars.temp[1] = IngresoPalabra("Apellido", (f + 1));  if (MisVars.temp[1] == " ") return;  MisVars.temp[2] = IngresoEdad(f + 1) ?? " ";  if (MisVars.temp[2] == " ") return;  for (int c = 0; c < 3; c++)  {  MisVars.personas[f, c] = MisVars.temp[c];  }  }  }  public static string IngresoPalabra(string titulo, int f)  {  int fails = 0;  do  {  Colores.ConsolaLetraColorYellow();  Console.Write("\nIngrese el " + titulo + " N°" + f + ": ");  Colores.ConsolaLetraColorWhite();  string nombre = (Console.ReadLine() ?? " ").Trim().ToLower();  if (Regex.IsMatch(nombre, @"^([a-z]){3,20}$"))  {  return nombre;  }  else  {  OpcionInvalida("Ingreso Invalido " + (++fails));  }  if (fails == 3)  {  OpcionInvalida("Demasiados intentos fallidos");  return null;  }  } while (fails < 3);  return null;  }  public static string IngresoEdad(int f)  {  int fails = 0;  do  {  Colores.ConsolaLetraColorYellow();  Console.Write("\nIngrese la Edad N°" + f + ": ");  Colores.ConsolaLetraColorWhite();  string edad = (Console.ReadLine() ?? " ").Trim().ToLower();  if (Regex.IsMatch(edad, @"^(\d){1,2}$"))  {  return edad;  }  else  {  OpcionInvalida("Ingreso Invalido " + (++fails));  }  if (fails == 3)  {  OpcionInvalida("Demasiados intentos fallidos");  return " ";  }  } while (fails < 3);  return " ";  }  }  }  **///////////////////////////////////////////////**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TPAnualEjer6  {  public static class MisVars  {  public static string[,] personas = new string[5, 3];  public static string[] temp = new string[3];  }  }  ////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TPAnualEjer6  {  public static class Colores  {  public static void ConsolaLetraColorRed() { Console.ForegroundColor = ConsoleColor.Red; }  public static void ConsolaLetraColorWhite() { Console.ForegroundColor = ConsoleColor.White; }  public static void ConsolaLetraColorMagenta() { Console.ForegroundColor = ConsoleColor.Magenta; }  public static void ConsolaLetraColorGreen() { Console.ForegroundColor = ConsoleColor.Green; }  public static void ConsolaLetraColorYellow() { Console.ForegroundColor = ConsoleColor.Yellow; }  public static void ConsolaLetraColorGray() { Console.ForegroundColor = ConsoleColor.Gray; }  }  }  ///////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Text.RegularExpressions;  namespace TPAnualEjer6  {  static class Program  {  private static void Main(string[] args)  {    Funs.App();  }      }  } |  |
| **Ejercicio 7:** Desarrollar un programa que simule que se arrojan 5 dados, 5 veces, sobre la mesa y se toma nota de los valores de cada dado (la cara que queda para arriba). Los valores deben ser cargados en un Array bidimensional siendo cada fila la que representa cada una de las 5 tiradas o tiros. Luego de estar cargado, el programa deberá recorrer el array e indicar cuántas veces salió cada cara del dado en total de todos los tiros. Se deberá respetar el siguiente formato: | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.IO;  namespace Ejercicio7  {  static class Program  {  public static void Main(string[] args)  {  Fun.App();  FunAccesoArchivo.CerrarTxt();  }      }  }  ///////////////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Ejercicio7  {  public static class Vars  {  public static int[,] dados = new int[5, 5];  public static int[] valor = new int[6];  public static string archivo = "historial.txt";  }  }  /////////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.IO;  namespace Ejercicio7  {  public static class AccesoArchivos  {  public static StreamWriter streamWriter = File.AppendText(Vars.archivo);  }  }  ///////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.IO;  namespace Ejercicio7  {  public static class Fun  {  public static void App()  {  string opcion;  do  {  Fun.MenuPrincipal();  opcion = Console.ReadLine() ?? "";  Fun.SeleccionOpcion(opcion);  } while (opcion != "0");  }  public static void SeleccionOpcion(string opcion)  {  Console.Clear();  switch (opcion)  {  case "1":  ResetValores();  TirarDados();  MostrarTiros();  MostrarValores();  break;  case "0":  return;  default:  OpcionInvalida("Opcion Invalida");  break;  }  }  public static void ResetValores()  {  for (int i = 0; i < 6; i++)  Vars.valor[i] = 0;  }  private static void TirarDados()  {  Random random = new Random();  for (int i = 0; i < 5; i++)  {  Console.WriteLine("Presione una tecla para la jugada numero " + (i + 1) + "...");  Console.ReadKey(true);  for (int j = 0; j < 5; j++)  {  Vars.dados[i, j] = random.Next(1, 7);  ContarValores(i, j);  }  }  }  private static void MostrarTiros()  {  Console.WriteLine("");  StringBuilder sb = new StringBuilder();  sb.Append("\n\*\*\*\n" + DateTime.Now.ToString());  for (int i = 0; i < 5; i++)  {  sb.Append("\nTiro " + (i + 1) + ": ");  Console.Write("Tiro " + (i + 1) + ": ");  for (int j = 0; j < 5; j++)  {  Console.Write(" " + Vars.dados[i, j] + " ");  sb.Append(" " + Vars.dados[i, j] + " ");  }  Console.WriteLine("\n");  }  }  private static void MostrarValores()  {  StringBuilder sb = new StringBuilder();  for (int i = 0; i < 6; i++)  {  Console.WriteLine(i + 1 + " = " + Vars.valor[i] + " veces");  sb.Append("\n" + i + 1 + " = " + Vars.valor[i] + " veces");  }    FunAccesoArchivo.Escribir(sb);    }  private static void ContarValores(int i, int j)  {  switch (Vars.dados[i, j])  {  case 1:  Vars.valor[0]++;  break;  case 2:  Vars.valor[1]++;  break;  case 3:  Vars.valor[2]++;  break;  case 4:  Vars.valor[3]++;  break;  case 5:  Vars.valor[4]++;  break;  case 6:  Vars.valor[5]++;  break;  default:  break;  }  }  public static void MenuPrincipal()  {  ConsolaLetraColorGreen();  Console.WriteLine("\nIngrese una opcion segun su indice y oprima enter\n");  ConsolaLetraColorWhite();  Console.WriteLine("1- Tirar Dados");  Console.WriteLine("0- Salir");  }  static void OpcionInvalida(string frase)  {  Console.Clear();  ConsolaLetraColorRed();  Console.WriteLine(frase + "\n");  }  static void ConsolaLetraColorRed() { Console.ForegroundColor = ConsoleColor.Red; }  static void ConsolaLetraColorWhite() { Console.ForegroundColor = ConsoleColor.White; }  static void ConsolaLetraColorGreen() { Console.ForegroundColor = ConsoleColor.Green; }  }  }  /////////////////////////////////////  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace Ejercicio7  {  public static class FunAccesoArchivo  {  public static void Escribir(StringBuilder frase)  {  AccesoArchivos.streamWriter.WriteLine(frase);  }  public static void CerrarTxt()  {  AccesoArchivos.streamWriter.Close();  }  }  } |  |
| **Ejercicio 9:** Desarrollar un juego (en lenguaje C# y en o de consola). El juego debe permitir que al menos dos jugadores puedan jugar de forma alternada y que al finalizar el mismo indique quien ganó. Pre-requisitos obligatorios: - Uso de funciones. - Menú recursivo. - Uso de clases globales. - Debe llevar un registro de los resultados finales en un archivo de texto. | |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class ElJuego**  **{**  **public static void Juego()**  **{**  **Program.JugadorUno.Reloj.Reset();**  **Program.JugadorDos.Reloj.Reset();**  **do**  **{**  **FunGl.VistaNormal();**  **FunGl.TurnoJugador();**  **string eleccion = FunGl.EleccionPosicion();**  **if (Program.ConfigVar.Sonido) Console.Beep();**  **if (eleccion == "0")**  **{**  **Console.Clear();**  **TeRendiste();**  **return;**  **}**  **bool marca = OpcionesJuego.MarcarPosicion(eleccion);**  **if (marca && Program.misVar.Jagadas > 4)**  **{**  **Program.misVar.Ganador = ElGanador.Ganador();**  **if (Program.misVar.Ganador)**  **{**  **PararRelojes();**  **ElGanador.MostrarGanador();**  **}**  **if (!Program.misVar.Ganador) Program.misVar.Ganador = Empate(Program.misVar.Ganador);**  **}**  **} while (!Program.misVar.Ganador);**  **Program.misVar.Ganador = false;**  **}**  **public static void PararRelojes()**  **{**  **Program.JugadorUno.Reloj.Stop();**  **Program.JugadorDos.Reloj.Stop();**  **}**  **public static void ResetPuntaje()**  **{**  **Program.JugadorUno.PuntoJugador = 0;**  **Program.JugadorDos.PuntoJugador = 0;**  **}**  **public static void TeRendiste()**  **{**  **if (Program.misVar.Turno) Console.WriteLine("Jugador UNO X Te rendiste...");**  **else Console.WriteLine("Jugador DOS O Te rendiste...");**  **ColorGl.ColorJugador();**  **FunGl.Mostrarcartel(Cartel.Ganador);**  **ElGanador.MiGanador();**  **ElTablero.TableroGanador();**  **FunGl.PressTecla();**  **}**  **public static bool Empate(bool ganador)**  **{**  **if (Program.misVar.Jagadas == 9 && !ganador)**  **{**  **Console.Clear();**  **FunGl.Mostrarcartel(Cartel.Tateti);**  **ColorGl.ConsolaLetraColorRed();**  **FunGl.Mostrarcartel(Cartel.Empate);**  **ElTablero.Tablero();**  **PFunAccesoArchivo.Escribir("EMPATE " + FunGl.PuntajeTiempoTxt());**  **FunGl.PressTecla();**  **return true;**  **}**  **return false;**  **}**  **}**  **}**  **///////////////////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class ElTablero**  **{**  **public static readonly char[,] posiciones = new char[3, 3];**  **public static void Tablero()**  **{**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("");**  **for (int i = 0; i < 3; i++)**  **{**  **Console.WriteLine("\t\t\t\t "+posiciones[i, 0]+" | "+posiciones[i, 1]+" | "+posiciones[i, 2]+"");**  **if (i < 2)**  **Console.WriteLine("\t\t\t\t ---------");**  **}**  **}**  **public static void TableroGanador()**  **{**  **Console.WriteLine("\n");**  **for (int i = 0; i < 3; i++)**  **{**  **Console.Write("\t\t\t\t\t ");**  **for (int j = 0; j < 3; j++)**  **{**  **ColorGl.LetraColor(posiciones[i, j]);**  **Console.Write(posiciones[i, j]+" ");**  **if (j < 2)**  **{**  **ColorGl.ConsolaLetraColorWhite();**  **Console.Write("| ");**  **}**  **}**  **if (i < 2)**  **{**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("\n\t\t\t\t\t ---------");**  **}**  **}**  **Console.WriteLine("");**  **}**  **public static void CargarPosiciones()**  **{**  **Program.misVar.Jagadas = 0;**  **int pos = 49;**  **for (int i = 0; i < 3; i++)**  **{**  **for (int j = 0; j < 3; j++)**  **posiciones[i, j] = (char)(j + pos);**  **pos += 3;**  **}**  **}**  **}**  **}**  **///////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class FunGl**  **{**  **public static void App()**  **{**  **string opcion;**  **do**  **{**  **ElTablero.CargarPosiciones();**  **FunGl.VistaNormal();**  **FunGl.MenuPrincipal();**  **opcion = Console.ReadLine() ?? "";**  **if (Program.ConfigVar.Sonido) Console.Beep();**  **OpcionesJuego.SeleccionOpcion(opcion);**  **} while (opcion != "0");**  **}**    **public static void SeleccionPuntajeTiempo()**  **{**  **if (!Program.misVar.Turno && Program.JugadorUno.Reloj.ElapsedMilliseconds < Program.JugadorDos.Reloj.ElapsedMilliseconds)**  **{**  **Program.JugadorUno.PuntoJugador++;**  **}**  **else if (Program.misVar.Turno && Program.JugadorDos.Reloj.ElapsedMilliseconds < Program.JugadorUno.Reloj.ElapsedMilliseconds)**  **{**  **Program.JugadorDos.PuntoJugador++;**  **}**  **}**  **public static void Mostrarcartel(string[] cartel)**  **{**  **ColorGl.ColorRandom();**  **foreach (var item in cartel)**  **{**  **Console.WriteLine(item);**  **}**  **}**  **public static string PuntajeTiempoTxt()**  **{**  **string puntaje = "\nPUNTAJE\n\"Jugador Uno X\": " + Program.JugadorUno.PuntoJugador + "\n\"Jugador Dos O\": " + Program.JugadorDos.PuntoJugador;**  **string tiempo = "\nTIEMPO\n\"Jugador Uno X\": " + Program.JugadorUno.Reloj.Elapsed + "\n\"Jugador Dos O\": " + Program.JugadorDos.Reloj.Elapsed;**  **return puntaje + tiempo;**  **}**  **public static void TurnoJugador()**  **{**  **if (Program.misVar.Turno)**  **{**  **MostrarJuagador("Uno X");**  **Program.JugadorDos.Reloj.Stop();**  **Program.JugadorUno.Reloj.Start();**  **}**  **else**  **{**  **MostrarJuagador("Dos O");**  **Program.JugadorUno.Reloj.Stop();**  **Program.JugadorDos.Reloj.Start();**  **}**  **}**  **public static void MostrarJuagador(string jugador)**  **{**  **ColorGl.ColorJugador();**  **Console.WriteLine(("\nTurno de jugador " + jugador).ToUpper());**  **}**  **public static void VistaNormal()**  **{**  **Console.Clear();**  **FunGl.Mostrarcartel(Cartel.Tateti);**  **MostrarPuntaje();**  **ElTablero.Tablero();**  **}**  **public static void MenuPrincipal()**  **{**  **ColorGl.ConsolaLetraColorYellow();**  **Console.WriteLine("\nIngrese una opcion segun su indice y oprima enter\n");**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("1- Jugar");**  **Console.WriteLine("2- Resetear Puntajes");**  **Console.WriteLine("3- Configuracion");**  **Console.WriteLine("0- Salir");**  **}**  **public static string EleccionPosicion()**  **{**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("\nElija la posicion del tablero disponible y oprima enter");**  **ColorGl.ConsolaLetraColorYellow();**  **Console.WriteLine("Presione 0 para Rendirse");**  **return Console.ReadLine() ?? " ";**  **}**  **public static void MostrarPuntaje()**  **{**  **ColorGl.ConsolaLetraColorYellow();**  **Console.WriteLine("\t\t\t\t PUNTAJE");**  **ColorGl.ConsolaLetraGreen();**  **Console.Write("\t \"Jugador Uno X\": " + Program.JugadorUno.PuntoJugador);**  **ColorGl.ConsolaLetraCyan();**  **Console.WriteLine("\t\t \"Jugador Dos O\": " + Program.JugadorDos.PuntoJugador);**  **}**  **public static void PressTecla()**  **{**  **Console.WriteLine("\nPresiona una tecla para continuar");**  **Console.ReadKey();**  **}**  **}**  **}**  **////////////////////////////////7**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Diagnostics;**  **namespace Ejercicio9**  **{**  **public struct JugadorStruct**  **{**  **public int PuntoJugador { get; set; } // pasar a structura jugador**  **public Stopwatch Reloj { get; set; } // pasar a structura jugador**  **}**  **}**  **//////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Diagnostics;**  **namespace Ejercicio9**  **{**  **public struct MisVar**  **{**  **public int Fails { get; set; } // estructura general**  **public Random NRandom { get; set; } // estructura general**  **public bool Turno { get; set; } // estructura general**  **public int Jagadas { get; set; } // estructura general**  **public bool Ganador { get; set; } // estructura general**  **public const int BasePuntaje = 11;**  **}**  **}**  **///////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Text.RegularExpressions;**  **namespace Ejercicio9**  **{**  **public static class OpcionesJuego**  **{**  **public static bool MarcarPosicion(string eleccion)**  **{**  **if (Regex.IsMatch(eleccion, @"^\d$"))**  **{**  **if (ValidarPosicion(eleccion))**  **{**  **Program.misVar.Turno = !Program.misVar.Turno;**  **Program.misVar.Fails = 0;**  **Program.misVar.Jagadas++;**  **return true;**  **}**  **else**  **{**  **OpcionInvalida("Opcion Invalida "+ Program.misVar.Fails);**  **}**  **}**  **else**  **{**  **OpcionInvalida("Opcion Invalida "+ Program.misVar.Fails);**  **}**  **return false;**  **}**  **public static bool ValidarPosicion(string eleccion)**  **{**  **char miEleccion = Convert.ToChar(eleccion);**  **for (int i = 0; i < 3; i++)**  **{**  **for (int j = 0; j < 3; j++)**  **{**  **if (ElTablero.posiciones[i, j] == miEleccion)**  **{**  **if (Program.misVar.Turno)**  **ElTablero.posiciones[i, j] = 'X';**  **else**  **ElTablero.posiciones[i, j] = 'O';**  **return true;**  **}**  **}**  **}**  **return false;**  **}**  **public static void SeleccionOpcion(string opcion)**  **{**  **Console.Clear();**  **switch (opcion)**  **{**  **case "1":**  **ElJuego.Juego();**  **break;**  **case "2":**  **ElJuego.ResetPuntaje();**  **break;**  **case "3":**  **Configuracion.MenuConfiguracion();**  **break;**  **case "0":**  **return;**  **default:**  **OpcionInvalida("Opcion Invalida");**  **break;**  **}**  **}**  **public static void OpcionInvalida(string frase)**  **{**  **Program.misVar.Fails++;**  **Console.Clear();**  **FunGl.Mostrarcartel(Cartel.Tateti);**  **ColorGl.ConsolaLetraColorRed();**  **Console.WriteLine(frase+"\n Presione una tecla para continuar");**  **Console.ReadKey(true);**  **}**  **}**  **}**  **/////////////////////////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.IO;**  **namespace Ejercicio9**  **{**  **public static class PAccesoArchivo**  **{**  **public static StreamWriter sw = File.AppendText(Program.ConfigVar.ArchivoHistorial);**    **}**  **}**  **///////////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class PFunAccesoArchivo**  **{**  **public static void Escribir(string frase)**  **{**  **PAccesoArchivo.sw.WriteLine("\n" + DateTime.Now + "\n" + frase);**  **PAccesoArchivo.sw.Flush();**  **}**  **public static void CerrarTxt()**  **{**  **PAccesoArchivo.sw.Close();**  **}**  **}**  **}**  **/////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Diagnostics;**  **using System.Threading;**  **namespace Ejercicio9**  **{**  **static class Program**  **{**    **public static MisVar misVar = new MisVar()**  **{**  **Fails = 0,**  **Jagadas = 0,**  **Turno = true,**  **Ganador = false,**  **NRandom = new Random()**  **};**  **public static ConfigStruct ConfigVar = new ConfigStruct()**  **{**  **Tema = true,**  **Sonido = false,**  **ArchivoHistorial = "historial.txt"**  **};**  **public static JugadorStruct JugadorUno = new JugadorStruct()**  **{**  **PuntoJugador = 0,**  **Reloj = new Stopwatch()**  **};**  **public static JugadorStruct JugadorDos = new JugadorStruct()**  **{**  **PuntoJugador = 0,**  **Reloj = new Stopwatch()**  **};**    **public static void Main()**  **{**  **Console.BackgroundColor = ConsoleColor.Black;**    **FunGl.App();**  **PFunAccesoArchivo.CerrarTxt();**  **}**  **}**  **}**  **//////////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class Cartel**  **{**  **public static string[] Tateti = new string[]**  **{ ".\_\_\_\_\_\_\_\_\_\_\_. \_\_\_ .\_\_\_\_\_\_\_\_\_\_\_. \_\_\_\_\_\_\_ .\_\_\_\_\_\_\_\_\_\_\_. \_\_ " ,**  **"| | / \\ | || \_\_\_\_| | || | " ,**  **"`---| |----´ / ^ \\ \_\_\_\_\_\_`---| |----´| |\_\_ \_\_\_\_\_\_`---| |----´| | " ,**  **" | | / /\_\\ \\ |\_\_\_\_\_\_| | | | \_\_| |\_\_\_\_\_\_| | | | | " ,**  **" | | / \_\_\_\_\_ \\ | | | |\_\_\_\_ | | | | ",**  **" |\_\_| /\_\_/ \\\_\_\\ |\_\_| |\_\_\_\_\_\_\_| |\_\_| |\_\_|"};**      **public static string[] Empate = new string[]**  **{ "\t \_\_\_\_\_\_\_ .\_\_\_ \_\_\_. .\_\_\_\_\_\_ \_\_\_ .\_\_\_\_\_\_\_\_\_\_\_. \_\_\_\_\_\_\_ " ,**  **"\t| \_\_\_\_|| \\/ | | \_ \\ / \\ | || \_\_\_\_|" ,**  **"\t| |\_\_ | \\ / | | |\_) | / ^ \\ `---| |----´| |\_\_ " ,**  **"\t| \_\_| | |\\/| | | \_\_\_/ / /\_\\ \\ | | | \_\_| " ,**  **"\t| |\_\_\_\_ | | | | | | / \_\_\_\_\_ \\ | | | |\_\_\_\_ " ,**  **"\t|\_\_\_\_\_\_\_||\_\_| |\_\_| | \_| /\_\_/ \\\_\_\\ |\_\_| |\_\_\_\_\_\_\_|" };**      **public static string[] Ganador = new string[]**  **{ "\t \_\_\_\_\_\_\_ \_\_\_ .\_\_ \_\_. \_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_ .\_\_\_\_\_\_ " ,**  **"\t / \_\_\_\_\_| / \\ | \\ | | / \\ | \\ / \_\_ \\ | \_ \\ " ,**  **"\t| | \_\_ / ^ \\ | \\| | / ^ \\ | .--. | | | | | |\_) | " ,**  **"\t| | |\_ | / /\_\\ \\ | . ` | / /\_\\ \\ | | | | | | | | / " ,**  **"\t| |\_\_| | / \_\_\_\_\_ \\ | |\\ | / \_\_\_\_\_ \\ | '--' | `--' | | |\\ \\----." ,**  **"\t \\\_\_\_\_\_\_| /\_\_/ \\\_\_\\ |\_\_| \\\_\_| /\_\_/ \\\_\_\\ |\_\_\_\_\_\_\_/ \\\_\_\_\_\_\_/ | \_| `.\_\_\_\_\_|" };**        **public static string[] JuadorX= new string[] {**  **" \_\_ \_\_ \_\_ \_\_\_\_\_\_\_ \_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_ .\_\_\_\_\_\_ \_\_\_ \_\_\_ " ,**  **" | | | | | | / \_\_\_\_\_| / \\ | \\ / \_\_ \\ | \_ \\ \\ \\ / / " ,**  **" | | | | | | | | \_\_ / ^ \\ | .--. | | | | | |\_) | \\ V / " ,**  **".--. | | | | | | | | |\_ | / /\_\\ \\ | | | | | | | | / > x < " ,**  **"| `--' | | `--' | | |\_\_| | / \_\_\_\_\_ \\ | '--' | `--' | | |\\ \\----. / . \\ " ,**  **" \\\_\_\_\_\_\_/ \\\_\_\_\_\_\_/ \\\_\_\_\_\_\_| /\_\_/ \\\_\_\\ |\_\_\_\_\_\_\_/ \\\_\_\_\_\_\_/ | \_| `.\_\_\_\_\_| /\_\_/ \\\_\_\\ " };**        **public static string[] JuadorO= new string[]**  **{ " \_\_ \_\_ \_\_ \_\_\_\_\_\_\_ \_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_ .\_\_\_\_\_\_ \_\_\_\_\_\_ " ,**  **" | | | | | | / \_\_\_\_\_| / \\ | \\ / \_\_ \\ | \_ \\ / \_\_ \\ " ,**  **" | | | | | | | | \_\_ / ^ \\ | .--. | | | | | |\_) | | | | | " ,**  **".--. | | | | | | | | |\_ | / /\_\\ \\ | | | | | | | | / | | | | " ,**  **"| `--' | | `--' | | |\_\_| | / \_\_\_\_\_ \\ | '--' | `--' | | |\\ \\----. | `--' | " ,**  **" \\\_\_\_\_\_\_/ \\\_\_\_\_\_\_/ \\\_\_\_\_\_\_| /\_\_/ \\\_\_\\ |\_\_\_\_\_\_\_/ \\\_\_\_\_\_\_/ | \_| `.\_\_\_\_\_| \\\_\_\_\_\_\_/ " };**        **}**  **}**  **///////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class ColorGl**  **{**  **public static void ColorJugador()**  **{**  **Console.ForegroundColor = Program.misVar.Turno ? ConsoleColor.DarkGreen : ConsoleColor.DarkCyan;**  **}**  **public static void LetraColor(char pos)**  **{**  **if (Program.misVar.Ganador && !Program.misVar.Turno && pos == 'X')**  **ConsolaLetraGreen();**  **else if (Program.misVar.Ganador && Program.misVar.Turno && pos == 'O')**  **ConsolaLetraCyan();**  **else**  **ConsolaLetraColorWhite();**  **}**  **public static void ColorRandom()**  **{**  **switch (Program.misVar.NRandom.Next(0, 5))**  **{**  **case 0:**  **ConsolaLetraGreen();**  **break;**  **case 1:**  **ConsolaLetraCyan();**  **break;**  **case 2:**  **ConsolaLetraColorRed();**  **break;**  **case 3:**  **ConsolaLetraColorMagenta();**  **break;**  **case 4:**  **ConsolaLetraColorYellow();**  **break;**  **default:**  **ConsolaLetraColorWhite();**  **break;**  **}**  **}**  **public static void ConsolaLetraGreen() { Console.ForegroundColor = ConsoleColor.DarkGreen;}**  **public static void ConsolaLetraCyan() { Console.ForegroundColor = ConsoleColor.DarkCyan;}**  **public static void ConsolaLetraColorRed() { Console.ForegroundColor = ConsoleColor.DarkRed;}**  **public static void ConsolaLetraColorMagenta() { Console.ForegroundColor = ConsoleColor.DarkMagenta;}**  **public static void ConsolaLetraColorYellow() { Console.ForegroundColor = ConsoleColor.DarkYellow; }**  **public static void ConsolaLetraColorWhite() {**  **if(Program.ConfigVar.Tema) Console.ForegroundColor = ConsoleColor.White;**  **else Console.ForegroundColor = ConsoleColor.Black;**  **}**  **}**  **}**  **///////////////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public struct ConfigStruct**  **{**  **public bool Sonido { get; set; } // estructura configuracion**  **public bool Tema { get; set; } // estructura configuracion**  **public string ArchivoHistorial { get; set; } // estructura configuracion**  **}**  **}**  **//////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class Configuracion**  **{**  **public static void MenuConfiguracion()**  **{**  **string opcion;**  **do**  **{**  **ElTablero.CargarPosiciones();**  **FunGl.VistaNormal();**  **Menu();**  **opcion = Console.ReadLine() ?? "";**  **if (Program.ConfigVar.Sonido) Console.Beep();**  **SeleccionOpcion(opcion);**  **} while (opcion != "0");**  **}**  **private static void Menu()**  **{**  **ColorGl.ConsolaLetraColorMagenta();**  **Console.WriteLine("\nIngrese una opcion segun su indice y oprima enter\n");**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("1- Menú Sonido");**  **Console.WriteLine("2- Menú Tema");**  **Console.WriteLine("0- Salir");**  **}**  **private static void SeleccionOpcion(string opcion)**  **{**  **Console.Clear();**  **switch (opcion)**  **{**  **case "1":**  **MenuSonido();**  **break;**  **case "2":**  **MenuTema();**  **break;**  **case "0":**  **return;**  **default:**  **OpcionesJuego.OpcionInvalida("Opcion Invalida");**  **break;**  **}**  **}**  **private static void MenuTema()**  **{**  **ColorGl.ConsolaLetraColorYellow();**  **Console.WriteLine("\nMenú TEMA\n");**  **Console.WriteLine("\nIngrese una opcion segun su indice y oprima enter\n");**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("1- Claro");**  **Console.WriteLine("2- Oscuro");**  **Console.WriteLine("0- Salir");**  **SeleccionTema();**  **}**  **private static void SeleccionTema()**  **{**  **string opcion = Console.ReadLine() ?? "2";**  **if (opcion != "1" && opcion != "2" && opcion != "0")**  **{**  **OpcionesJuego.OpcionInvalida("Opcion Invalidad");**  **}**  **else if (opcion == "1")**  **{**  **Program.ConfigVar.Tema = false;**  **Console.BackgroundColor = ConsoleColor.Gray;**  **}**  **else if (opcion == "2")**  **{**  **Program.ConfigVar.Tema = true;**  **Console.BackgroundColor = ConsoleColor.Black;**  **}**    **}**  **private static void MenuSonido()**  **{**  **ColorGl.ConsolaLetraColorYellow();**  **Console.WriteLine("\nMenú SONIDO\n");**  **Console.WriteLine("\nIngrese una opcion segun su indice y oprima enter\n");**  **ColorGl.ConsolaLetraColorWhite();**  **Console.WriteLine("1- ON");**  **Console.WriteLine("2- OFF");**  **Console.WriteLine("0- Salir");**  **SeleccionSonido();**    **}**  **private static void SeleccionSonido()**  **{**  **string opcion = Console.ReadLine() ?? "2";**  **if (opcion != "1" && opcion != "2" && opcion != "0")**  **{**  **OpcionesJuego.OpcionInvalida("Opcion Invalidad");**  **}**  **else**  **{**  **Program.ConfigVar.Sonido = opcion == "1" ;**  **if(Program.ConfigVar.Sonido) Console.Beep();**  **}**  **}**  **}**  **}**  **/////////////////////////////**  **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **namespace Ejercicio9**  **{**  **public static class ElGanador**  **{**  **public static void MostrarGanador()**  **{**  **Console.Clear();**  **FunGl.Mostrarcartel(Cartel.Ganador);**  **MiGanador();**  **ElTablero.TableroGanador();**  **FunGl.PressTecla();**  **}**  **public static void MiGanador()**  **{**  **string frase = "El ganador es el jugador ";**  **if (!Program.misVar.Turno)**  **{**  **ColorGl.ConsolaLetraGreen();**  **FunGl.Mostrarcartel(Cartel.JuadorX);**  **frase += "UNO X";**  **//FunGl.SeleccionPuntaje();**  **Program.JugadorUno.PuntoJugador += (MisVar.BasePuntaje - Program.misVar.Jagadas);**  **FunGl.SeleccionPuntajeTiempo();**  **}**  **else**  **{**  **ColorGl.ConsolaLetraCyan();**  **frase += "DOS O";**  **FunGl.Mostrarcartel(Cartel.JuadorO);**  **//FunGl.SeleccionPuntaje();**  **Program.JugadorDos.PuntoJugador += (MisVar.BasePuntaje - Program.misVar.Jagadas);**  **FunGl.SeleccionPuntajeTiempo();**  **}**  **PFunAccesoArchivo.Escribir(frase + FunGl.PuntajeTiempoTxt());**  **}**  **public static bool Ganador() { return GanadorHorizontal() || GanadorVertical() || GanadorDiagonal(); }**  **public static bool GanadorDiagonal()**  **{**  **if (ElTablero.posiciones[0, 0] == ElTablero.posiciones[1, 1] && ElTablero.posiciones[0, 0] == ElTablero.posiciones[2, 2])**  **return true;**  **if (ElTablero.posiciones[2, 0] == ElTablero.posiciones[1, 1] && ElTablero.posiciones[2, 0] == ElTablero.posiciones[0, 2])**  **return true;**  **return false;**  **}**  **public static bool GanadorVertical()**  **{**  **for (int i = 0; i < 3; i++)**  **{**  **if (ElTablero.posiciones[0, i] == ElTablero.posiciones[1, i] && ElTablero.posiciones[0, i] == ElTablero.posiciones[2, i])**  **return true;**  **}**  **return false;**  **}**  **public static bool GanadorHorizontal()**  **{**  **for (int i = 0; i < 3; i++)**  **{**  **if (ElTablero.posiciones[i, 0] == ElTablero.posiciones[i, 1] && ElTablero.posiciones[i, 0] == ElTablero.posiciones[i, 2])**  **return true;**  **}**  **return false;**  **}**  **}**  **}** |  |