Contenido

[RANDOM 3](#_Toc66733454)

[LEER FILAS DE UNA TABLA 3](#_Toc66733455)

[TRANSFORMA UN STRING EN UN CHARARRAY 3](#_Toc66733456)

[MANEJO DE CHAR 4](#_Toc66733457)

[.ReadKey() 4](#_Toc66733458)

[.KeyChar 4](#_Toc66733459)

[Ejemplo 4](#_Toc66733460)

[ENUM 4](#_Toc66733461)

[ConsoleColor 4](#_Toc66733462)

[StringBuilder 5](#_Toc66733463)

[Example: Insert() 5](#_Toc66733464)

[Example: Remove() 5](#_Toc66733465)

[Example: Replace() 6](#_Toc66733466)

[Points to Remember 6](#_Toc66733467)

[CONSOLA 6](#_Toc66733468)

[Ubicar Cursor 6](#_Toc66733469)

[Borra pantalla 6](#_Toc66733470)

[Sleep 6](#_Toc66733471)

[Colores 6](#_Toc66733472)

[LISTAS 7](#_Toc66733473)

[FILES 7](#_Toc66733474)

[REPETIR CADENAS O CARACTERES 7](#_Toc66733475)

[Cadenas 7](#_Toc66733476)

[Caracteres 8](#_Toc66733477)

[Helper 9](#_Toc66733478)

[Ingreso string 9](#_Toc66733479)

[Ingreso Int 9](#_Toc66733480)

[Ingreso Double 10](#_Toc66733481)

[Ingreso decimal 10](#_Toc66733482)

[SEGUNDO PARCIAL 12](#_Toc66733483)

[Gustavo 12](#_Toc66733484)

[Pablo 12](#_Toc66733485)

[EXAMEN DICIEMBRE 14](#_Toc66733486)

[RESOLUCIÓN 14](#_Toc66733487)

[Program.cs 14](#_Toc66733488)

[Helpers.cs 15](#_Toc66733489)

[EXAMEN DICIEMBRE – PRODUCTOS DESDE ARCHIVO DE TEXTO 16](#_Toc66733490)

[Program.cs 16](#_Toc66733491)

[ENCABEZADO DE LISTA GENERICA 18](#_Toc66733492)

[EXAMEN DICIEMBRE RESUELTO OPTIMIZADO (MENOS CODIGO, CON METODOS PARA UBICAR LAS CABECERAS. 19](#_Toc66733493)

[PROGRAM 19](#_Toc66733494)

[HELPERS 20](#_Toc66733495)

[EXAMEN DE DICIEMBRE – OPTIMIZADO 2 (CON RESIZE DE ARRAYS PARA AGREGAR PRODUCTOS) 22](#_Toc66733496)

[Program 22](#_Toc66733497)

[Helpers 23](#_Toc66733498)

[EXAMEN FINAL MARZO 2021 24](#_Toc66733499)

# RANDOM

private int CalcularPuntaje()

{

var random = new Random();

return random.Next(1, 10); // devuelve num aleatorio >= 1 y <10

}

//

// Resumen:

// Devuelve un entero aleatorio que está dentro de un intervalo especificado.

//

// Parámetros:

// minValue:

// Devuelve el límite inferior inclusivo del número aleatorio.

//

// maxValue:

// Devuelve el límite superior exclusivo del número aleatorio. maxValuedebe ser

// mayor o igual que minValue.

//

// Devuelve:

// Un entero de 32 bits con signo mayor o igual que minValue y menor que maxValue;

// es decir, que incluye el intervalo de valores devueltos minValue pero no maxValue.

// Si minValue es igual a maxValue, minValue se devuelve.

//

// Excepciones:

// T:System.ArgumentOutOfRangeException:

// minValue es mayor que maxValue.

public virtual int Next(int minValue, int maxValue);

# LEER FILAS DE UNA TABLA

public void MostrarPuntajeJugador()

{

var miBLL = new BusinessLogicLayer();

var tabla = miBLL.ObtenerPuntajes(NombreJugador);

for (var i = 0; i < tabla.Rows.Count; i++)

{

var puntaje = tabla.Rows[i]["puntaje"];

var fecha = tabla.Rows[i]["FechaUltimoAcceso"];

Console.WriteLine($"Fecha:{fecha} Puntaje:{puntaje}");

}

}

# TRANSFORMA UN STRING EN UN CHARARRAY

char[] \_palabraAhorcado = "AHORCADO".ToCharArray();

# MANEJO DE CHAR

## .ReadKey()

//

// Resumen:

// Obtiene la siguiente tecla de carácter o de función presionada por el usuario.

// La tecla presionada se muestra en la ventana de la consola.

//

// Devuelve:

// Objeto que describe la constante System.ConsoleKey y el carácter Unicode, si

// existe, que corresponden a la tecla presionada en la consola. El objeto System.ConsoleKeyInfo

// también describe, en una combinación bit a bit de valores de System.ConsoleModifiers,

// si alguna de las teclas modificadoras Mayús, Alt o Ctrl se presionaron al mismo

// tiempo que la tecla de la consola.

//

// Excepciones:

// T:System.InvalidOperationException:

// La propiedad System.Console.In se ha redirigido desde alguna otra secuencia distinta

// de la consola.

public static ConsoleKeyInfo ReadKey();

## .KeyChar

//

// Resumen:

// Obtiene el carácter Unicode al que el objeto System.ConsoleKeyInfo actual representa.

//

// Devuelve:

// Objeto correspondiente a la tecla de consola a la que el objeto System.ConsoleKeyInfo

// representa.

public char KeyChar { get; }

## Ejemplo

Console.WriteLine("Ingrese un caracter: ");

var letra = Console.ReadKey().KeyChar;

letra = char.ToUpper(letra);

# ENUM

Genero una lista de constantes a las que puedo asignarle valores enteros y que pueden ser devueltos en otra parte del programa con el método🡪 .Basico (devuelve 5) o .Seccion3 (devuelve 9)

public enum Nivel : int

{

Basico = 5,

Medio = 6,

Avanzado = 8,

SuperHeavy = 10

}

private enum PosTop : int

{

Seccion1 = 1,

Seccion2 = 5,

Seccion3 = 9

}

## ConsoleColor

namespace System

{

//

// Resumen:

// Especifica las constantes que definen los colores de primer plano y de fondo

// de la consola.

public enum ConsoleColor

{

Black = 0,

DarkBlue = 1,

DarkGreen = 2,

DarkCyan = 3,

DarkRed = 4,

DarkMagenta = 5,

DarkYellow = 6,

Gray = 7,

DarkGray = 8,

Blue = 9,

Green = 10,

Cyan = 11,

Red = 12,

Magenta = 13,

Yellow = 14,

White = 15

}

# StringBuilder

StringBuilder sb = new StringBuilder(); //construye el objeto sb tipo StringBuilder

sb.Append("Hello "); //agrega Hello a sb

sb.AppendLine("World!"); // agrega World! a sb y un salto de línea al final.

sb.AppendLine("Hello C#"); // agrega Hello C# a sb y un salto de línea al final.

Console.WriteLine(sb); //muestra sb (“Hello World! En una línea y Hello C# en otra)

sbAmout.AppendFormat("{0:C} ", 25); // al objeto sbAmount le da formato Currency (devuelve $25.00)

## Example: Insert()

StringBuilder sb = new StringBuilder("Hello World!");

sb.Insert(5," C#");

Console.WriteLine(sb); //output: Hello C# World!

## Example: Remove()

StringBuilder sb = new StringBuilder("Hello World!!",50);

sb.Remove(6, 2);

Console.WriteLine(sb); //output: Hello rld!!

## Example: Replace()

StringBuilder sb = new StringBuilder("Hello World!");

sb.Replace("World", "C#");

Console.WriteLine(sb);//output: Hello C#!

Points to Remember :

1. StringBuilder is mutable.
2. StringBuilder performs faster than string when appending multiple string values.
3. Use StringBuilder when you need to append more than three or four strings.
4. Use the Append() method to add or append strings to the StringBuilder object.
5. Use the ToString() method to retrieve a string from the StringBuilder object.

# CONSOLA

## Ubicar Cursor

private void UbicarCursor(PosTop pos) /// método definido por usuario enum definido previamente parámetro recibido

{

var posTop = (int)pos;

Console.SetCursorPosition(1, posTop); //coloca el cursor en columna 1, fila postop

}

Console.CursorLeft; //devuelve la posición de la columna en la que se encuentra el cursor

Console.CursorTop // devuelve la posición de la fila del cursor

Console.SetCursorPosition(posLeft, posTop) // Establece la posición del cursor en posLeft,posTop

## Borra pantalla

Console.Clear(); //borra la pantalla

## Sleep

Thread.Sleep(250); // inserta un retraso de 250 (mseg?)

## Colores

Console.BackgroundColor = (ConsoleColor)random.Next(0, 16); // asigna un color de fondo a lo escrito

# LISTAS

List<int> intList1 = new List<int>();

intList1.Add(100);

intList1.Add(2);

intList1.Add(30);

intList1.Add(40);

List<int> intList2 = new List<int>();

intList2.Add(50);

intList2.AddRange(intList1);

//Console.WriteLine(intList2.Count);

int total = intList2.Count;

intList2.Sort();

for (int i=0;i<total;i++)

{

Console.WriteLine(intList2[i]);

}

# FILES

Var líneas = File.ReadAllLines("Palabras.txt") // lee el archivo de texto y devuelve una matriz de Strings donde cada elemento es una línea del archivo de texto.

Línea = random.Next(1, lineas.Length) // devuelve un numero al azar entre 1 y la cantidad de líneas que tiene el archivo de texto.

linea = lineas[numLinea];//leo la linea en la posición aleatoria

linea.Trim();//borramos los espacios en blanco al principio y al final

# REPETIR CADENAS O CARACTERES

## Cadenas

string.Join("", Enumerable.Repeat("ab", 2)); // returns abab

## Caracteres

new string('a', 2); // returns aa

# Helper

## Ingreso string

public static string IngresoString(string mensaje)

{

string ingreso = null;

while (string.IsNullOrWhiteSpace(ingreso))

{

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return ingreso;

}

## Ingreso Int

public static int IngresoInt()

{

int salida;

salida = IngresoInt("El valor ingresado no es numérico. Ingrese nuevamente");

return salida;

}

public static int IngresoInt(string mensaje)

{

int salida = IngresoInt(int.MinValue, int.MaxValue, mensaje); //$"Ingrese un valor Numérico");

return salida;

}

public static int IngresoInt(int valordesde, int valorHasta, string mensaje)

{

int salida = IngresoInt(valordesde, valorHasta, mensaje, false);

return salida;

}

public static int IngresoInt(int valordesde, int valorHasta)

{

return IngresoInt(valordesde, valorHasta, $"Ingrese un valor entre {valordesde} y {valorHasta}");

}

public static int IngresoInt(int valordesde, int valorHasta, string mensaje, bool limpiarPantalla)

{

int salida;

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

string ingreso = Console.ReadLine();

while (int.TryParse(ingreso, out salida) == false || (salida <= valordesde || salida >= valorHasta))

{

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return salida;

}

## Ingreso Double

public static double IngresoDouble()

{

double salida;

salida = IngresoDouble("Ingrese un número");

return salida;

}

public static double IngresoDouble(string mensaje)

{

double salida;

salida = IngresoDouble(double.MinValue, double.MaxValue, mensaje);

return salida;

}

public static double IngresoDouble(double valordesde, double valorHasta)

{

double salida;

salida = IngresoDouble(valordesde, valordesde, $"Ingrese un valor entre {valordesde} y {valorHasta}");

return salida;

}

public static double IngresoDouble(double valordesde, double valorHasta, string mensaje)

{

double salida;

salida = IngresoDouble(valordesde, valorHasta, mensaje, false);

return salida;

}

public static double IngresoDouble(double valordesde, double valorHasta, string mensaje, bool limpiarPantalla)

{

double salida;

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

string ingreso = Console.ReadLine();

while (double.TryParse(ingreso, out salida) == false || (salida < valordesde || salida > valorHasta))

{

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return salida;

}

## Ingreso decimal

public static decimal IngresoDecimal()

{

decimal salida;

salida = IngresoDecimal("Ingrese un número");

return salida;

}

public static decimal IngresoDecimal(string mensaje)

{

decimal salida;

salida = IngresoDecimal(decimal.MinValue, decimal.MaxValue, mensaje);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta)

{

decimal salida;

salida = IngresoDecimal(valordesde, valordesde, $"Ingrese un valor entre {valordesde} y {valorHasta}");

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje)

{

decimal salida;

salida = IngresoDecimal(valordesde, valorHasta, mensaje, false);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje, bool limpiarPantalla)

{

decimal salida;

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

string ingreso = Console.ReadLine();

while (decimal.TryParse(ingreso, out salida) == false || (salida < valordesde || salida > valorHasta))

{

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return salida;

}

# SEGUNDO PARCIAL

## Gustavo

class Program

{

static void Main(string[] args)

{

Examen.Examen.Empezar();//NO BORRAR ESTA LINEA

//ESCRIBE TU CODIGO AQUI:

////////////Primera consigna

string [] palabras = File.ReadAllLines("Palabras.txt");

Console.WriteLine("Las palabras ubicadas en el rango de posiciones de la 5 a la 30 son:\n");

for (int i = 5; i < 31; i++)

{

Console.WriteLine(palabras[i]);

}

Console.WriteLine("Presione una tecla para continuar");

Console.ReadLine();

////////////Segunda consigna

Console.WriteLine();

Console.WriteLine($"La cantidad de elementos en el vector palabras es: {palabras.Length}\n");

Console.WriteLine("Presione una tecla para continuar");

Console.ReadLine();

////////////Tercera consigna

string ingresoLetra;

Console.WriteLine("Ingrese: \npara mostrar el nombre ---> Y o y " +

"\npara limpiar la pantalla y mostrar mensaje ---> Q o q \n" +

"o cualquier otra letra para finalizar");

ingresoLetra = Console.ReadLine().ToString().ToUpper();

if (ingresoLetra == "Y")

{

Console.Clear();

Console.SetCursorPosition(25, 20);

Console.WriteLine("Gustavo Omar Furrer");

}

else

{

if (ingresoLetra == "Q")

{

Console.Clear();

Console.BackgroundColor = ConsoleColor.Blue;

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("Esto ha sido el parcial");

}

else

{

Console.Beep(8000, 1000);

}

}

//Console.ReadLine();

Examen.Examen.Finalizar();//NO BORRAR ESTA LINEA

}

}

## Pablo

# EXAMEN DICIEMBRE

## RESOLUCIÓN

### Program.cs

class Program

{

static void Main(string[] args)

{

int tamanioVector = 2;

decimal ganancia = 0.15m;

decimal iva = 0.21m;

string[] productos = new string[tamanioVector];

decimal[] precioCosto = new decimal[tamanioVector];

decimal[] precioVenta = new decimal[tamanioVector];

decimal[] montoIva = new decimal[tamanioVector];

decimal[] precioFinal = new decimal[tamanioVector];

for (int i = 0; i < tamanioVector; i++)

{

var ingresoNombre = Helpers.IngresoString("Ingrese el nombre del producto");

productos[i] = ingresoNombre;

var ingresoCosto = Helpers.IngresoDecimal(0, decimal.MaxValue, "Ingrese el precio de costo", true);

precioCosto[i] = ingresoCosto;

Console.Clear();

}

for (int i = 0; i < tamanioVector; i++)

{

precioVenta[i] = precioCosto[i] \* (1 + ganancia);

montoIva[i] = precioVenta[i] \* iva;

precioFinal[i] = Math.Round(precioVenta[i] + montoIva[i], 2);

}

Console.WriteLine("=====================================================================================================");

Console.WriteLine("LISTA DE PRECIOS");

Console.WriteLine("=====================================================================================================");

Console.WriteLine($"Producto: Precio Venta IVA Precio final");

Console.WriteLine("-----------------------------------------------------------------------------------------------------");

for (int i = 0; i < tamanioVector; i++)

{

var posTop = 6;

var digitosPV = precioVenta[i].ToString().Length;

var digitosI = montoIva[i].ToString().Length;

var digitosPF = precioFinal[i].ToString().Length;

var posLeft = 52;

Console.SetCursorPosition(0, posTop + i);

Console.WriteLine(productos[i]);

Console.SetCursorPosition(posLeft-digitosPV, posTop + i);

Console.WriteLine($"{ precioVenta[i]:c2}");

Console.SetCursorPosition(posLeft - digitosI + 21, posTop + i);

Console.WriteLine($"{ montoIva[i]:c2}");

Console.SetCursorPosition(posLeft-digitosPF + 43, posTop + i);

Console.WriteLine($"{ precioFinal[i]:c2}");

//Console.WriteLine($"Producto: {productos[i]} - Precio Venta: {precioVenta[i]:c2} - IVA: {montoIva[i]:C2}" +

// $" - Precio final: {precioFinal[i]:c2}");

//Console.WriteLine("-----------------------------------------------------------------------------------------------------");

}

Console.WriteLine("-----------------------------------------------------------------------------------------------------");

Console.WriteLine($"Cantidad de productos: {productos.Length}");

Console.ReadLine();

}

}

### Helpers.cs

class Helpers

{

public static string IngresoString(string mensaje)

{

string ingreso = null;

while (string.IsNullOrWhiteSpace(ingreso))

{

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

Console.Clear();

}

return ingreso;

}

public static decimal IngresoDecimal()

{

decimal salida;

salida = IngresoDecimal("Ingrese un número");

return salida;

}

public static decimal IngresoDecimal(string mensaje)

{

decimal salida;

salida = IngresoDecimal(decimal.MinValue, decimal.MaxValue, mensaje);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta)

{

decimal salida;

salida = IngresoDecimal(valordesde, valordesde, $"Ingrese un valor entre {valordesde} y {valorHasta}");

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje)

{

decimal salida;

salida = IngresoDecimal(valordesde, valorHasta, mensaje, false);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje, bool limpiarPantalla)

{

decimal salida;

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

string ingreso = Console.ReadLine();

while (decimal.TryParse(ingreso, out salida) == false || (salida < valordesde || salida > valorHasta))

{

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return salida;

}

}

## EXAMEN DICIEMBRE – PRODUCTOS DESDE ARCHIVO DE TEXTO

### Program.cs

class Program

{

static void Main(string[] args)

{

int tamanioVector = 10;

decimal ganancia = 0.15m;

decimal iva = 0.21m;

string[] productos = new string[tamanioVector];

decimal[] precioCosto = new decimal[tamanioVector];

decimal[] precioVenta = new decimal[tamanioVector];

decimal[] montoIva = new decimal[tamanioVector];

decimal[] precioFinal = new decimal[tamanioVector];

for (int i = 0; i < tamanioVector; i++)

{

string linea;

var texto = File.ReadAllLines("Productos.txt");

Random random = new Random();

do

{

var numLinea = random.Next(0, 10);

linea = texto[numLinea];

productos[i] = (productos.Contains(linea)) ? productos[i] = null : productos[i] = linea;

} while (productos[i] == null);

var mensaje = $"Ingrese el precio del producto: {linea}";

var ingresoCosto = Helpers.IngresoDecimal(0, decimal.MaxValue, mensaje, true);

precioCosto[i] = ingresoCosto;

Console.Clear();

}

for (int i = 0; i < tamanioVector; i++)

{

precioVenta[i] = precioCosto[i] \* (1 + ganancia);

montoIva[i] = precioVenta[i] \* iva;

precioFinal[i] = Math.Round(precioVenta[i] + montoIva[i], 2);

}

Console.WriteLine("=====================================================================================================");

Console.WriteLine("LISTA DE PRECIOS");

Console.WriteLine("=====================================================================================================");

Console.WriteLine($"Producto: Precio Venta IVA Precio final");

Console.WriteLine("-----------------------------------------------------------------------------------------------------");

for (int i = 0; i < tamanioVector; i++)

{

var posTop = 6;

var digitosPV = precioVenta[i].ToString().Length;

var digitosI = montoIva[i].ToString().Length;

var digitosPF = precioFinal[i].ToString().Length;

var posLeft = 52;

Console.SetCursorPosition(0, posTop + i);

Console.WriteLine(productos[i]);

Console.SetCursorPosition(posLeft - digitosPV, posTop + i);

Console.WriteLine($"{ precioVenta[i]:c2}");

Console.SetCursorPosition(posLeft - digitosI + 21, posTop + i);

Console.WriteLine($"{ montoIva[i]:c2}");

Console.SetCursorPosition(posLeft - digitosPF + 43, posTop + i);

Console.WriteLine($"{ precioFinal[i]:c2}");

//Console.WriteLine($"Producto: {productos[i]} - Precio Venta: {precioVenta[i]:c2} - IVA: {montoIva[i]:C2}" +

// $" - Precio final: {precioFinal[i]:c2}");

//Console.WriteLine("-----------------------------------------------------------------------------------------------------");

}

Console.WriteLine("-----------------------------------------------------------------------------------------------------");

Console.WriteLine($"Cantidad de productos: {productos.Length}");

Console.ReadLine();

}

}

## ENCABEZADO DE LISTA GENERICA

/// ENCABEZADO DE LISTA GENERICA

int ancho = 90;

string titulo = "LISTA DE PRECIOS";

string guiones = new string('-',ancho);

string dobleLinea = new string('=', ancho);

string cabecera1 = ("Producto").PadRight(40);

string cabecera2 = ("Precio Venta").PadLeft(20);

string cabecera3 = ("IVA").PadLeft(10);

string cabecera4 = ("Precio Final").PadLeft(20);

Console.WriteLine(guiones);

Console.WriteLine((titulo.PadLeft((ancho + Math.Abs(titulo.Length)) / 2).PadRight(ancho)));

Console.WriteLine(dobleLinea);

Console.WriteLine(cabecera1 + cabecera2+ cabecera3 + cabecera4);

Console.ReadLine();

## EXAMEN DICIEMBRE RESUELTO OPTIMIZADO (MENOS CODIGO, CON METODOS PARA UBICAR LAS CABECERAS.

### PROGRAM

class Program

{

static void Main(string[] args)

{

int tamanioVector = 2;

decimal ganancia = 0.15m;

decimal iva = 0.21m;

string[] productos = new string[tamanioVector];

decimal[] precioCosto = new decimal[tamanioVector];

decimal[] precioVenta = new decimal[tamanioVector];

decimal[] montoIva = new decimal[tamanioVector];

decimal[] precioFinal = new decimal[tamanioVector];

int ancho = 90;

string titulo = "LISTA DE PRECIOS";

string guiones = new string('-', ancho);

string dobleLinea = new string('=', ancho);

string cabecera1 = Helpers.Cabecera("Producto", 38,'d');

string cabecera2 = Helpers.Cabecera("Precio Venta",22);

string cabecera3 = Helpers.Cabecera("IVA", 12);

string cabecera4 = Helpers.Cabecera("Precio Final", 18);

for (int i = 0; i < tamanioVector; i++)

{

var ingresoNombre = Helpers.IngresoString("Ingrese el nombre del producto");

productos[i] = ingresoNombre;

var ingresoCosto = Helpers.IngresoDecimal(0, decimal.MaxValue, "Ingrese el precio de costo", true);

precioCosto[i] = ingresoCosto;

Console.Clear();

}

for (int i = 0; i < tamanioVector; i++)

{

precioVenta[i] = precioCosto[i] \* (1 + ganancia);

montoIva[i] = precioVenta[i] \* iva;

precioFinal[i] = Math.Round(precioVenta[i] + montoIva[i], 2);

}

Console.WriteLine(dobleLinea);

Console.WriteLine((titulo.PadLeft((ancho + Math.Abs(titulo.Length)) / 2).PadRight(ancho)));

Console.WriteLine(dobleLinea);

Console.WriteLine(cabecera1 + cabecera2 + cabecera3 + cabecera4);

Console.WriteLine((guiones));

for (int i = 0; i < tamanioVector; i++)

{

Console.Write( Helpers.Cabecera(productos[i], 39, 'd'));

Console.Write(Helpers.Cabecera($"{ precioVenta[i]:c2}", 19));

Console.Write(Helpers.Cabecera($"{ montoIva[i]:c2}", 15));

Console.WriteLine(Helpers.Cabecera($"{ precioFinal[i]:c2}", 17));

}

Console.WriteLine(guiones);

Console.WriteLine($"Cantidad de productos: {productos.Length}");

Console.WriteLine("Presione enter para finalizar");

Console.ReadLine();

}

}

### HELPERS

class Helpers

{

public static string Cabecera(string mensaje, int posicion, char lado)

{

//string Cabe = (lado == 'd') ? mensaje.PadRight(posicion) : mensaje.PadLeft (posicion);

string Cabe = mensaje.PadRight(posicion);

return Cabe;

}

public static string Cabecera(string mensaje, int posicion)

{

string Cabe = mensaje.PadLeft(posicion); //Cabecera(mensaje, posicion,'l');

return Cabe;

}

public static string IngresoString(string mensaje)

{

string ingreso = null;

while (string.IsNullOrWhiteSpace(ingreso))

{

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

Console.Clear();

}

return ingreso;

}

public static decimal IngresoDecimal()

{

decimal salida;

salida = IngresoDecimal("Ingrese un número");

return salida;

}

public static decimal IngresoDecimal(string mensaje)

{

decimal salida;

salida = IngresoDecimal(decimal.MinValue, decimal.MaxValue, mensaje);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta)

{

decimal salida;

salida = IngresoDecimal(valordesde, valordesde, $"Ingrese un valor entre {valordesde} y {valorHasta}");

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje)

{

decimal salida;

salida = IngresoDecimal(valordesde, valorHasta, mensaje, false);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje, bool limpiarPantalla)

{

decimal salida;

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

string ingreso = Console.ReadLine();

while (decimal.TryParse(ingreso, out salida) == false || (salida < valordesde || salida > valorHasta))

{

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return salida;

}

}

## EXAMEN DE DICIEMBRE – OPTIMIZADO 2 (CON RESIZE DE ARRAYS PARA AGREGAR PRODUCTOS)

### Program

class Program

{

static void Main(string[] args)

{

char salir = 'n';

int tamanioVector = 2;

decimal ganancia = 0.15m;

decimal iva = 0.21m;

string[] productos = new string[tamanioVector];

decimal[] precioCosto = new decimal[tamanioVector];

decimal[] precioVenta = new decimal[tamanioVector];

decimal[] montoIva = new decimal[tamanioVector];

decimal[] precioFinal = new decimal[tamanioVector];

int cont = 0;

do

{

Console.Clear();

int ancho = 90;

string titulo = "LISTA DE PRECIOS";

string guiones = new string('-', ancho);

string dobleLinea = new string('=', ancho);

string cabecera1 = Helpers.Cabecera("Producto", 38, 'd');

string cabecera2 = Helpers.Cabecera("Precio Venta", 22);

string cabecera3 = Helpers.Cabecera("IVA", 12);

string cabecera4 = Helpers.Cabecera("Precio Final", 18);

for (int i = cont; i < tamanioVector; i++)

{

var ingresoNombre = Helpers.IngresoString("Ingrese el nombre del producto");

productos[i] = ingresoNombre;

var ingresoCosto = Helpers.IngresoDecimal(0, decimal.MaxValue, "Ingrese el precio de costo", true);

precioCosto[i] = ingresoCosto;

Console.Clear();

}

for (int i =cont; i < tamanioVector; i++)

{

precioVenta[i] = precioCosto[i] \* (1 + ganancia);

montoIva[i] = precioVenta[i] \* iva;

precioFinal[i] = Math.Round(precioVenta[i] + montoIva[i], 2);

}

Console.WriteLine(dobleLinea);

Console.WriteLine((titulo.PadLeft((ancho + Math.Abs(titulo.Length)) / 2).PadRight(ancho)));

Console.WriteLine(dobleLinea);

Console.WriteLine(cabecera1 + cabecera2 + cabecera3 + cabecera4);

Console.WriteLine((guiones));

for (int i = 0; i < tamanioVector; i++)

{

Console.Write(Helpers.Cabecera(productos[i], 39, 'd'));

Console.Write(Helpers.Cabecera($"{ precioVenta[i]:c2}", 19));

Console.Write(Helpers.Cabecera($"{ montoIva[i]:c2}", 15));

Console.WriteLine(Helpers.Cabecera($"{ precioFinal[i]:c2}", 17));

}

Console.WriteLine(guiones);

Console.WriteLine($"Cantidad de productos: {productos.Length}");

Console.WriteLine("Presione s para salir, u otra tecla para agregar un producto");

salir = Console.ReadKey().KeyChar;

salir = char.ToUpper(salir);

Array.Resize<string>(ref productos, tamanioVector + 1);

Array.Resize<decimal>(ref precioCosto, tamanioVector + 1);

Array.Resize<decimal>(ref precioVenta, tamanioVector + 1);

Array.Resize<decimal>(ref montoIva, tamanioVector + 1);

Array.Resize<decimal>(ref precioFinal, tamanioVector + 1);

cont = tamanioVector;

tamanioVector++;

} while (salir != 'S');

//Console.ReadLine();

}

}

### Helpers

class Helpers

{

public static string Cabecera(string mensaje, int posicion, char lado)

{

//string Cabe = (lado == 'd') ? mensaje.PadRight(posicion) : mensaje.PadLeft (posicion);

string Cabe = mensaje.PadRight(posicion);

return Cabe;

}

public static string Cabecera(string mensaje, int posicion)

{

string Cabe = mensaje.PadLeft(posicion); //Cabecera(mensaje, posicion,'l');

return Cabe;

}

public static string IngresoString(string mensaje)

{

string ingreso = null;

while (string.IsNullOrWhiteSpace(ingreso))

{

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

Console.Clear();

}

return ingreso;

}

public static decimal IngresoDecimal()

{

decimal salida;

salida = IngresoDecimal("Ingrese un número");

return salida;

}

public static decimal IngresoDecimal(string mensaje)

{

decimal salida;

salida = IngresoDecimal(decimal.MinValue, decimal.MaxValue, mensaje);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta)

{

decimal salida;

salida = IngresoDecimal(valordesde, valordesde, $"Ingrese un valor entre {valordesde} y {valorHasta}");

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje)

{

decimal salida;

salida = IngresoDecimal(valordesde, valorHasta, mensaje, false);

return salida;

}

public static decimal IngresoDecimal(decimal valordesde, decimal valorHasta, string mensaje, bool limpiarPantalla)

{

decimal salida;

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

string ingreso = Console.ReadLine();

while (decimal.TryParse(ingreso, out salida) == false || (salida < valordesde || salida > valorHasta))

{

if (limpiarPantalla == true)

{

Console.Clear();

}

Console.WriteLine(mensaje);

ingreso = Console.ReadLine();

}

return salida;

}

}

# EXAMEN FINAL MARZO 2021

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ManuelBelgrano.Examen

{

    class Program

    {

        static void Main(string[] args)

        {

            var tamanioVector = 10;

            string[] alumnos = new string[tamanioVector];

            double[] nota1erExamen = new double[tamanioVector];

            double[] nota2doExamen = new double[tamanioVector];

            double[] nota3erExamen = new double[tamanioVector];

            double[] promedio = new double[tamanioVector];

            string[] condicion = new string[tamanioVector];

            for (int i = 0; i < tamanioVector; i++)

            {

                alumnos[i] = Helper.IngresoString("Ingrese el nombre del Alumno:");

                nota1erExamen[i] = Helper.IngresoDouble(0,10,"Ingrese la nota del 1er examen");

                nota2doExamen[i] = Helper.IngresoDouble(0, 10, "Ingrese la nota del 2do examen");

                nota3erExamen[i] = Helper.IngresoDouble(0, 10, "Ingrese la nota del 3er examen");

                Console.Clear();

            }

            for (int i = 0; i < tamanioVector; i++)

            {

                promedio[i] = Math.Round(((nota1erExamen[i] + nota2doExamen[i]) + nota3erExamen[i]) / 3,2);

                condicion[i] = (promedio[i] >= 6) ? condicion[i] = "Aprobado" : condicion[i] = "Reprobado";

            }

            for (int i = 0; i < tamanioVector; i++)

            {

                Console.WriteLine($"Alumno: {alumnos[i]} , Primer Examen: {nota1erExamen[i]} , Segundo Examen: {nota2doExamen[i]} , Tercer Examen: {nota3erExamen[i]} ," +

                    $" Promedio: {promedio[i]} , Condición: {condicion[i]}");

                Console.WriteLine();

            }

            Console.ReadLine();

        }

    }

}

