C# | Documentos para o 2º Teste

## Ficha 05 - Arrays Bidimensionais

**1.** Uma fábrica possui 3 linhas de produção, cada uma delas com um conjunto de 4 máquinas. Escreve um programa que permita ler a produção de cada máquina (nas 3 linhas de produção) e determinar:

a) a máquina com menor produção

b) o total produzido nas diversas linhas de produção

using System;

Esquema básico de cabeçalho quando se inicia um novo projecto de C# no Visual Studio.

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Threading;

namespace Exercicios

{

class Program

{

**static void Main(string[] args)**

**{**

**// Ficha 05 - Questão 01**

int linhaProducao = 0;

int maquina = 0;

int producaoMin = int.MaxValue;

int soma = 0;

Console.WriteLine("MATRIZ 3x4\n");

int[,] matriz = new int[3, 4] { { 0, 0, 0, 0 }, { 0, 0, 0, 0 }, { 0, 0, 0, 0 } };

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 4; j++)

{

Console.Write("Máquina {0} da {1}ª linha de produção: ", j + 1, i + 1);

matriz[i, j] = Convert.ToInt16(Console.ReadLine());

soma = soma + matriz[i, j];

if (matriz[i, j] < producaoMin)

{

producaoMin = matriz[i, j];

linhaProducao = i+1;

maquina = j+1;

}

}

Console.WriteLine("\n");

}

//----------------------------- Como obter o Min() num array two-dimensional -----------------------

int menorProducao = matriz.Cast<int>().Min();

// http://stackoverflow.com/questions/3049240/obtaining-the-min-and-max-of-a-two-dimensional-array-using-linq

//-----------------------------------------------------------------------------------------------------------

Console.WriteLine("A máquina com menor produção foi a máquina {0} da {1}ª linha de produção: ",maquina,linhaProducao);

Console.WriteLine("A produção da máquina foi de {0} unidades.", producaoMin);

Console.WriteLine("A produção total = {0}", soma);

// Console.ReadLine();

} // Main()

}

**}**

**2.** Dada uma matriz 3x3 de inteiros, cujos valores são indicados pelo utilizador, elabore um programa

que imprima a sua transposta.

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

**// Ficha 05 - Questão 02 | Receber dados num array e fazer a sua transposta**

int[,] matriz = new int[3,3]; // declaração de uma matriz bidimensional

Console.WriteLine("Introduza os valores da matriz (3x3): ");

for (int i = 0; i <= 2; i++) // percorrer linhas

{

for (int j = 0; j <= 2; j++) // percorrer colunas

{

Console.SetCursorPosition(j \* 7, 5 + i \* 2); // variar a posição da linha e coluna

matriz[i, j] = Convert.ToInt16(Console.ReadLine());

// Console.Clear();

}

// OBS: Setposition precende sempre um readline ou writeline

}

Console.WriteLine("\n\nMATRZ TRANSPOSTA ");

for (int i = 0; i <= 2; i++) // percorrer linhas

{

Variação da Posição

j => variação para a direita

i => variação para baixo

for (int j = 0; j <= 2; j++) // percorrer colunas

{

Console.SetCursorPosition(j \* 7, 15 + i \* 2); // variar a posição da linha e coluna

Console.WriteLine(matriz[j, i]);

}

}

Console.ReadKey();

}

}

}

**3.** Altere o programa anterior, de forma a que o tamanho da matriz (N = nº de linhas e

de M = nº de colunas) seja indicado pelo utilizador, e o seu preenchimento seja

efetuado com valores aleatórios, compreendidos entre 1 e N\*M.

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

// 03 C# - Matriz n por m e números aleatórios

Console.WriteLine("Indique a ordem da matriz: ");

int ordem = Convert.ToInt16(Console.ReadLine());

//Console.WriteLine("Indique o numero de colunas da matriz: ");

//int coluna = Convert.ToInt16(Console.ReadLine());

int[,] matriz = new int[ordem, ordem]; // declaração de uma matriz bidimensional

Random numero\_randomico = new Random(); // declaração da variável randomica

Console.WriteLine("\n\nMatriz ({0}x{1}): ", ordem, ordem);

int temp = 0;

for (int i = 0; i < ordem; i++) // percorrer linhas

{

for (int j = 0; j < ordem; j++) // percorrer colunas

{

matriz[i, j] = numero\_randomico.Next(1, ordem \* ordem);

Console.SetCursorPosition(j \* 7, 6 + i \* 2); // variar a posição da linha e coluna

Console.WriteLine(matriz[i, j]);

temp = 5 + i \* 2;

}

// OBS: Setposition precende sempre um readline ou writeline

}

Console.WriteLine("\n\nTransposta da Matriz ");

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

{

for (int j = 0; j < ordem; j++)

{

Console.SetCursorPosition(j \* 7, (temp + 6) + i \* 2);

Console.WriteLine(matriz[j, i]);

}

}

Console.WriteLine();

//Console.ReadKey();

}

}

}

**4.** Elabore um programa com o seguinte menu inicial:

a) Inicializar uma matriz quadrada (consiste em pedir ao utilizador as dimensões

da matriz e introduzir valores aleatórios);

b) Imprimir a matriz transposta;

c) Calcular o valor máximo da matriz e indicar a sua posição (linha, coluna);

d) Sair do programa (termina o programa).

**using System.Threading;**

namespace ConsoleApplication1

{

class Program

{

**// 04 C# - Menu com opcões sobre Matriz** | Realizei o exercício com prática em variáveis globais, permitindo operar sempre sobre uma mesma matriz.

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// VARIAVEIS GLOBAIS \*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Global

{

public static int ordem; // vai estipular a ordem da matriz

public static int[,] matriz; // declaração da matriz

public static int index\_i; // capturar o indice i do valor Max

public static int index\_j; // capturar o indice j do valor Max

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \* OPCAO 1 \*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

static void criarMatriz()

{

Console.Clear();

int maximo = int.MinValue;

Console.WriteLine("Indique a ordem da matriz: ");

Global.ordem = Convert.ToInt16(Console.ReadLine());

Global.matriz = new int[Global.ordem, Global.ordem]; // declaração de uma matriz bidimensional

Random numero\_randomico = new Random(); // declaração da variável randomica

Console.WriteLine("\n\nMatriz ({0}x{1}): ", Global.ordem, Global.ordem);

for (int i = 0; i < Global.ordem; i++) // percorrer linhas

{

for (int j = 0; j < Global.ordem; j++) // percorrer colunas

{

Global.matriz[i, j] = numero\_randomico.Next(1, Global.ordem \* Global.ordem);

if (Global.matriz[i, j] > maximo)

{

maximo = Global.matriz[i, j];

Global.index\_i = i;

Global.index\_j = j;

}

Console.SetCursorPosition(38 + j \* 7, 5 + i \* 2); // variar a posição da linha e coluna

Console.WriteLine(Global.matriz[i, j]);

}

// OBS: Setposition precende sempre um readline ou writeline

}

Console.WriteLine("Tecle ENTER para volar");

Console.ReadLine();

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \* OPCAO 2 \*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

static void transpostaDaMatriz(int[,] matriz)

{

Console.Clear();

for (int i = 0; i < Global.ordem; i++) // percorrer linhas

{

for (int j = 0; j < Global.ordem; j++) // percorrer colunas

{

Console.SetCursorPosition(38 + j \* 7, 5 + i \* 2); // variar a posição da linha e coluna

Console.WriteLine(matriz[j, i]);

}

// OBS: Setposition precende sempre um readline ou writeline

}

Console.WriteLine("Tecle ENTER para volar");

Console.ReadLine();

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \* OPCAO 3 \*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

static void valorMaximo()

{

int max = Global.matriz.Cast<int>().Max();

Console.Clear();

Console.WriteLine("O valor márixo da matriz é " + max);

Console.WriteLine("A posição do valor máximo é a linha {0} e coluna {1}. ", Global.index\_i + 1, Global.index\_j + 1);

Console.WriteLine("\nTecle ENTER para volar");

Console.ReadLine();

}

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// \* Main() -> Programa prinicipal \***

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**static void Main(string[] args)**

{

int opcao;

do

{

Console.Clear();

Console.WriteLine(

"\n+++ PROGRAMA MATRIZ +++\n" +

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" +

"\nOPCOES DO MENU\n\n" +

"1 - Inicializar uma matriz quadrada\n" +

"2 - Imprimir a matriz transposta\n" +

"3 - Determinar o valor máximo\n" +

"4 - Sair do programa\n\n" +

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

);

opcao = Convert.ToInt16(Console.ReadLine());

switch (opcao)

{

case 1:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* OPCAO 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

criarMatriz();

break;

case 2:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* OPCAO 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

transpostaDaMatriz(Global.matriz);

break;

case 3:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* OPCAO 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

valorMaximo();

break;

case 4: // Não faz nada e sai do programa, pois é a condição de saída.

break;

default:

Console.SetCursorPosition(0, 10);

Console.WriteLine("::MENSAGEM:: Opção inválida");

Thread.Sleep(500);

// Thread.Sleep(500);

//Console.ReadLine();

break;

}

} while (opcao != 4);

// Console.ReadKey();

}

}

} **// Fim da questão 04 da ficha 05 - Arrays Bidimensionais**

## Ficha 06 – Strings

**// 01. Elabore um algoritmo que leia uma frase e escreva os seus caracteres por ordem inversa.**

namespace Strings

{

class Program

{

static void Main(string[] args)

{

String frase = "";

Console.WriteLine("Escreva uma frase: ");

frase = Console.ReadLine();

int comprimento\_da\_frase = frase.Length;

for (int i = comprimento\_da\_frase - 1

; i >= 0

; i--)

{

Console.Write(frase[i]);

}

Console.WriteLine("\n");

}

}

}

**Ficha 06 – Strings // 02. Lê uma frase e retorna número de caracteres, número de espaços, número de vogais**

namespace Strings

{

class Program

{

static void Main(string[] args)

{

String frase = "";

int conta\_espacos = 0;

int conta\_vogais = 0;

Console.Write("Escreva uma frase: ");

frase = Console.ReadLine();

// Número de caracteres:

int comprimento\_da\_frase = frase.Length;

// Número de espaços

for (int i = 0; i < comprimento\_da\_frase; i++)

{

if (frase[i] == ' ')

conta\_espacos += 1;

}

frase = frase.ToLower();

// Número de vogais

for (int i = 0; i < comprimento\_da\_frase; i++)

{

if (frase[i] == 'a' ||

frase[i] == 'e' ||

frase[i] == 'i' ||

frase[i] == 'o' ||

frase[i] == 'u')

conta\_vogais += 1;

}

// Respostas

Console.Write("\nNúmero de Caracteres: {0} ", comprimento\_da\_frase);

Console.Write("\nNúmero de espaços: {0} ", conta\_espacos);

Console.Write("\nNúmero de vogais: {0} ", conta\_vogais);

Console.WriteLine("\n");

Console.ReadLine();

}

}

}

**Ficha 06 – Strings // 03.** Escreva um algoritmo que leia um texto e valide se é capicua ou não. Uma capicua consiste num texto que tanto pode ser lido da esquerda para a direita como da direita para a esquerda. Exemplos de capicuas: osso, asa, ana

namespace Strings

{

class Program

{

static void Main(string[] args)

{

String texto = " ";

int comp = 0;

bool flag = true;

Console.Write("Escreva uma frase: ");

texto = Console.ReadLine();

// Comprimento do texto

comp = texto.Length;

int j = comp - 1;

for (int i = 0; i <= comp / 2; i++)

{

if (texto[i] != texto[j])

flag = false;

else

j--;

}

if (flag == true)

Console.WriteLine("É uma capicua.");

else

Console.WriteLine("Não é uma capicua.");

Console.WriteLine("\n");

Console.ReadLine();

}

}

}

**Ficha 06 – Strings // 04 - Substituir as sequências de dois ou mais espaços em branco**

**// Pretende-se que se remova quando houver 2 ou mais espaços**

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

string frase = " ";

//string nova\_frase = " ";

Console.WriteLine("Insira uma frase com vários espaços entre as palavras: ");

frase = Console.ReadLine();

int comp = frase.Length;

// Retirando os espaços em branco extras da frase

for (int i = 0; i < comp - 1; i++)

{

if (frase[i] == ' ')

while (frase[i + 1] == ' ')

{

frase = frase.Remove(i, 1);

comp--; // compensar o tamanho do comprimento

// a cada vez que remove um espaço em branco,

// deminuímos também uma unidade do comprimento

}

}

Console.WriteLine(frase);

Console.ReadLine();

}

}

}

**Ficha 06 – Strings // 05 - Primeiro e último nome** | O programa recebe um nome completo e devolve apenas o primeiro e último nomes

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

String nome = " ";

String prenome = " ";

String sobrenome = " ";

Console.WriteLine("Digite o nome completo: ");

nome = Console.ReadLine();

// Comprimento do nome completo

int comp = nome.Length;

// Captura o primeiro nome

prenome = nome.Substring(0, nome.IndexOf(' '));

// Captura o último nome

int posicao\_ultimo\_espaco = nome.LastIndexOf(' ');

sobrenome = nome.Substring(posicao\_ultimo\_espaco, comp - posicao\_ultimo\_espaco);

// Apenas para teste, apresento o nome inicialmente recebido

Console.WriteLine("\n\nO nome completo era: \n{0} ", nome);

// Apresentação do nome simplificado

Console.WriteLine(prenome + sobrenome);

Console.ReadLine();

}

}

}

**Ficha 06 – Strings // 06 - Primeiro e último nome com os outros nomes abreviados**

// O programa recebe um nome completo e devolve o primeiro, os nomes do meio abreviados e último nome

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

String nome = " ";

String prenome = " ";

String sobrenome = " ";

Console.WriteLine("Digite o nome completo: ");

nome = Console.ReadLine();

// Comprimento do nome completo

int comp = nome.Length;

// Captura o primeiro nome

prenome = nome.Substring(0, nome.IndexOf(' '));

// Captura o último nome

int posicao\_ultimo\_espaco = nome.LastIndexOf(' ');

sobrenome = nome.Substring(posicao\_ultimo\_espaco + 1, comp - posicao\_ultimo\_espaco - 1);

// + 1 para pegar uma casa a frente do espaço

// - 1 para compensar e não estrapolar

/// Iimprime o primeiro nome

Console.Write(prenome + ' ');

// Busca iniciais do meio do nome

for (int i = 0; i < nome.LastIndexOf(' '); i++)

{

if (nome[i] == ' ')

{

Console.Write(nome[i + 1] + ". ");

}

}

// Iimprime o último nome

Console.Write(sobrenome);

Console.ReadLine();

}

}

}

**Ficha 06 – Strings // 07 - Dado um texto, determina a posição (no texto) e o número de ocorrências de determinada palavra.**

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

string texto = "Através de uma unidade de GPS/GPRS instalada em cada veículo, "

+ "\num aparelho de monitorização regista informação em tempo real "

+ "\nsobre a localização de cada veículo (latitude a longitude), assim como "

+ "\noutros dados complementares: velocidade instantânea, distancia percorrida "

+ "\ndesde o último registo, combustível em depósito, etc.";

string termoPesquisa = "";

string indiceTermo = "";

int contador = 0;

Console.WriteLine(texto);

//string textoLower = texto.ToLower;

Console.Write("\nDado o texto acima, escolha um termo a pesquisar: ");

termoPesquisa = Console.ReadLine();

for (int i = 0; i <= texto.Length - termoPesquisa.Length; i++)

{

if (texto.Substring(i, termoPesquisa.Length) == termoPesquisa)

{

contador++;

indiceTermo = indiceTermo + Convert.ToInt16(i + 1) + ", ";

}

}

// Retirar última vírgula e espaço (subtrair os dois últimos caracteres)

if (contador >= 1)

indiceTermo = indiceTermo.Substring(0, indiceTermo.Length - 2);

else

indiceTermo = "Nenhum termo para contar o índice..."; // se não encontrar nenhum termo igual

// Resposta final

Console.WriteLine("Nº de ocorrências: {0}\nPosições no texto: {1}", contador, indiceTermo);

}

}

}

**Ficha 06 – Strings // 08 - Converte um número decimal para numeração romana. Até 3 dígitos**

// http://www.climaat.angra.uac.pt/produtos/calculadoras/numeros\_romanos.htm

// https://pt.wikipedia.org/wiki/Numera%C3%A7%C3%A3o\_romana

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

String numero = " ";

String[] unidades = new String[10] { "", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX" };

String[] dezenas = new String[10] { "", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC" };

String[] centenas = new String[10] { "", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM" };

String[] guarda\_numeros = new String[3]; // irá guardar o resultado finao em ordem inversa

int comp = 0; // irá receber o comprimento do número digitado

// irá repetir até que o utilizador entre apenas com um número até 3 dígitos

do

{

// Recebe o número em decimal na forma de String

Console.WriteLine("Digite um número decimal: ");

numero = Console.ReadLine();

// Comprimento do número

comp = numero.Length;

// Converter o número que está em String para um Inteiro

int numero\_inteiro = Convert.ToInt16(numero);

// Processo de conversão

int temp = 0; // irá receber a unidade do número

int aux = 0; // irá multiplicar o número que está em temp por 1, 10 ou 100

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

{

temp = numero\_inteiro % 10; // recebe a unidade do número

numero\_inteiro = numero\_inteiro / 10; // reduz o número retirando a unidade

aux = temp \* Convert.ToInt16(Math.Pow(10, i)); // multiplica o número que está em temp por 1, 10 ou 100

// a potência devolve sempre um número em double

if (aux < 10)

{

guarda\_numeros[i] = unidades[aux]; // vai buscar o equivalente em romanos

}

else if (aux < 100)

{

aux = aux / 10; // volta a deixar somente o número mais significativo

guarda\_numeros[i] = dezenas[aux]; // vai buscar o equivalente em romanos

}

else if (aux < 1000)

{

aux = aux / 100; // volta a deixar somente o número mais significativo

guarda\_numeros[i] = centenas[aux]; // vai buscar o equivalente em romanos

}

}

} while (comp > 3); // explicação no início do ciclo "do .. while"

// Apresentar o resultado final

// Correr o array "guarda\_numeros" de trás para frente,

// uma vez que os números foram salvos em ordem contrária

Console.Write("Conversão do número {0} para Romanos: ", numero);

for (int i = comp - 1; i >= 0; i--)

{

Console.Write(guarda\_numeros[i]);

}

Console.ReadLine();

}

}

}

### Ficha 07 | Ficheiros te Texto

**01 - // Exercícios sobre Ficheiros**

// Faz o registo de **Número**, **Nome** e **Turma**

namespace Exercicios

{

class Program

{

**//------------------------**

**// 1 - REGISTAR UM ALUNO**

**//------------------------**

static void registoAluno()

{

// Imprime os atributos a ler

Console.Clear();

Console.SetCursorPosition(5, 2);

Console.WriteLine("CADASTRO DE NOVO ALUNO\n\n");

Console.SetCursorPosition(5, 4);

Console.WriteLine("Número : ");

Console.SetCursorPosition(5, 5);

Console.WriteLine("Nome : ");

Console.SetCursorPosition(5, 6);

Console.WriteLine("Turma : ");

// Leitura dos atributos e criação das variáveis: número, nome e turma

Console.SetCursorPosition(15, 4);

int numero = int.Parse(Console.ReadLine());

Console.SetCursorPosition(15, 5);

string nome = Console.ReadLine();

Console.SetCursorPosition(15, 6);

string turma = Console.ReadLine();

// Grava dados em ficheiro

// Criação do ficheiro

string ficheiroRegisto = @"alunos.txt";

StreamWriter registar;

if (File.Exists(ficheiroRegisto) == true)

{

registar = File.AppendText(ficheiroRegisto);

}

else

{

registar = File.CreateText(ficheiroRegisto);

}

// Grava no ficheiro

string linha = numero.ToString() + ";" + nome + ";" + turma;

registar.WriteLine(linha);

registar.Close();

}

**//------------------------**

**// 2 - CONSULTA GERAL**

**//------------------------**

static void consultaGeralAlunos()

{

Console.Clear();

int lin = 5;

string ficheiroRegisto = @"alunos.txt";

StreamReader consultar;

// verifica se o ficheiro existe

if (File.Exists(ficheiroRegisto) == true)

{

Console.SetCursorPosition(5, lin - 3); Console.WriteLine("-----------------------------------------");

Console.SetCursorPosition(5, lin - 2); Console.WriteLine("NUMERO\tNOME\t\t\tTURMA");

Console.SetCursorPosition(5, lin - 1); Console.WriteLine("-----------------------------------------");

consultar = File.OpenText(ficheiroRegisto); // abre o ficheiro para leitura

string linha = "";

while ((linha = consultar.ReadLine()) != null) // Lê ficheiro até ao final

{

int pos = linha.IndexOf(";");

string num = linha.Substring(0, pos); //Obtem o 1º campo de cada linha

// Refaz a variavel linha. a partir do 1º ";" até ao final da linha

linha = linha.Substring(pos + 1, linha.Length - pos - 1);

pos = linha.IndexOf(";");

string nome = linha.Substring(0, pos);

string turma = linha.Substring(pos + 1, linha.Length - pos - 1);

// Immprime os dados

Console.SetCursorPosition(5, lin);

Console.WriteLine(num);

Console.SetCursorPosition(16, lin);

Console.WriteLine(nome);

Console.SetCursorPosition(40, lin);

Console.WriteLine(turma);

lin++;

}

consultar.Close(); // fecha o ficheiro

}

else

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("\* MENSAGEM \* » Não existe nenhum ficheiro de registo.");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

Console.ReadLine();

}

**//----------------------------------**

**// 3 - CONSULTA ALUNOS POR TURMA**

**//----------------------------------**

// Nesta opção, é pedida qual a turma que deseja consultar,

// e então é listada todos os alunos desta mesma turma

static void consultaAlunosTurma()

{

Console.Clear();

int lin = 5;

string ficheiroRegisto = @"alunos.txt";

StreamReader consultar;

bool existeRegisto = false;

// verifica se o ficheiro existe

if (File.Exists(ficheiroRegisto) == true)

{

// Pede a turma que o utilizador deseja listar

Console.Write("Turma? ");

//char turmaProcurada = Convert.ToChar(Console.ReadLine());

string turmaProcurada = Console.ReadLine();

Console.SetCursorPosition(5, lin - 3); Console.WriteLine("-----------------------------------------");

Console.SetCursorPosition(5, lin - 2); Console.WriteLine("NUMERO\tNOME\t\t\tTURMA");

Console.SetCursorPosition(5, lin - 1); Console.WriteLine("-----------------------------------------");

consultar = File.OpenText(ficheiroRegisto); // abre o ficheiro para leitura

string linha = "";

while ((linha = consultar.ReadLine()) != null) // Lê ficheiro até ao final

{

int pos = linha.IndexOf(";");

if (linha.Substring

(linha.LastIndexOf(";")+1, (linha.Length -1) - (linha.LastIndexOf(";")))

== turmaProcurada)

// if (linha[linha.Length - 1] == turmaProcurada) // outra opcao para caso de turma com uma letra

{

string num = linha.Substring(0, pos); //Obtem o 1º campo de cada linha

// Refaz a variavel linha. a partir do 1º ";" até ao final da linha

linha = linha.Substring(pos + 1, linha.Length - pos - 1);

pos = linha.IndexOf(";");

string nome = linha.Substring(0, pos);

string turma = linha.Substring(pos + 1, linha.Length - pos - 1);

// Immprime os dados

Console.SetCursorPosition(5, lin);

Console.WriteLine(num);

Console.SetCursorPosition(16, lin);

Console.WriteLine(nome);

Console.SetCursorPosition(40, lin);

Console.WriteLine(turma);

lin++;

existeRegisto = true;

}

}

consultar.Close(); // fecha o ficheiro

// Mensagem caso não tenha nenhum registo da turma pesuqisada

if (existeRegisto == false)

{

Console.SetCursorPosition(5, lin); Console.WriteLine("Nao há registos desta turma \"{0}\"", turmaProcurada);

}

}

else

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("\* MENSAGEM \* » Não existe nenhum ficheiro de registo.");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

Console.ReadLine();

}

**// Programa Principal**

static void Main(string[] args)

{

char opcao = ' ';

do

{

Console.Clear();

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

Console.WriteLine("\tMENU");

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

Console.WriteLine("1 - Criar Registo de Aluno");

Console.WriteLine("2 - Consultar lista geral de alunos");

Console.WriteLine("3 - Consultar alunos por turma");

Console.WriteLine("0 - Sair");

Console.Write("\nOpção: ");

opcao = Convert.ToChar(Console.ReadLine());

switch (opcao)

{

case '1':

**Ficheiro => alunos.txt**

**Resultado no ficheiro de texto:**

789;Rita;A

987;Dante;A

888;Bia;B

456;Carlos;B

654;Stella;B

555;Leo;A

999;Dantiii;abc

registoAluno(); // feito

break;

case '2':

consultaGeralAlunos(); // feito

break;

case '3':

consultaAlunosTurma(); // feito

break;

default:

if (opcao != '0')

Console.WriteLine("Opção inválida!");

else

Console.WriteLine("Saindo do programa... <\"ENTER\" para sair>");

break;

}

} while (opcao != '0');

// Console.ReadLine();

}

}

}

### Ficha 08 | Ficheiros (também)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Exercicios

{

class Program

{

**// Ficha 08 | 01** - **Cadastro Aluno** com **Encriptação** e **Desencrip** e opções de **Alterar** e **Deletar** um registo

**//------------------------**

**// 1 - REGISTAR UM ALUNO**

**//------------------------**

static void registoAluno()

{

// Imprime os atributos a ler

Console.Clear();

Console.SetCursorPosition(5, 2);

Console.WriteLine("CADASTRO DE NOVO ALUNO\n\n");

Console.SetCursorPosition(5, 4);

Console.WriteLine("Número : ");

Console.SetCursorPosition(5, 5);

Console.WriteLine("Nome : ");

Console.SetCursorPosition(5, 6);

Console.WriteLine("Turma : ");

// Leitura dos atributos e criação das variáveis: número, nome e turma

Console.SetCursorPosition(15, 4);

int numero = int.Parse(Console.ReadLine());

Console.SetCursorPosition(15, 5);

string nome = Console.ReadLine();

Console.SetCursorPosition(15, 6);

string turma = Console.ReadLine();

// Grava dados em ficheiro

// Criação do ficheiro

string ficheiroRegisto = @"alunos\_encript.txt";

StreamWriter registar; // StreamWriter ou StreamReader faz a ponte de ligação entre o ficheiro e os comandos do programa

if (File.Exists(ficheiroRegisto) == true)

{

registar = File.AppendText(ficheiroRegisto);

}

else

{

registar = File.CreateText(ficheiroRegisto);

}

// Grava no ficheiro

string linha = numero.ToString() + ";" + nome + ";" + turma;

linha = encriptar(linha); // ENCRIPTAR A LINHA PARA O FICHEIRO >> chamada para a função encriptar()

registar.WriteLine(linha);

registar.Close();

}

**//------------------------**

**// ENCRIPTAR**

**//------------------------**

static string encriptar(string linha)

{

string novaLinha = "";

for (int i = 0; i < linha.Length; i++)

{

int code = Convert.ToInt16(linha[i]);

code = code + 3;

novaLinha = novaLinha + Convert.ToChar(code); // 3 é uma chave

}

return novaLinha;

}

**//------------------------**

**// DESENCRIPTAR**

**//------------------------**

static string desencriptar(string linha)

{

string novaLinha = "";

for (int i = 0; i < linha.Length; i++)

{

int code = Convert.ToInt16(linha[i]);

code = code - 3;

novaLinha = novaLinha + Convert.ToChar(code); // 3 é uma chave

}

return novaLinha;

}

**//------------------------**

**// 2 - CONSULTA GERAL**

**//------------------------**

static void consultaGeralAlunos()

{

Console.Clear();

int lin = 5;

string ficheiroRegisto = @"alunos\_encript.txt";

StreamReader consultar;

// verifica se o ficheiro existe

if (File.Exists(ficheiroRegisto) == true)

{

Console.SetCursorPosition(5, lin - 3); Console.WriteLine("-----------------------------------------");

Console.SetCursorPosition(5, lin - 2); Console.WriteLine("NUMERO\tNOME\t\t\tTURMA");

Console.SetCursorPosition(5, lin - 1); Console.WriteLine("-----------------------------------------");

consultar = File.OpenText(ficheiroRegisto); // abre o ficheiro para leitura

string linha = "";

while ((linha = consultar.ReadLine()) != null) // Lê ficheiro até ao final

{

linha = desencriptar(linha); // DESENCRIPTAR A LINHA DO FICHEIRO

int pos = linha.IndexOf(";");

string num = linha.Substring(0, pos); // Obtem o 1º campo de cada linha

// Refaz a variável linha a partir do 1º ";" até ao final da linha

linha = linha.Substring(pos + 1, linha.Length - pos - 1);

pos = linha.IndexOf(";");

string nome = linha.Substring(0, pos);

string turma = linha.Substring(pos + 1, linha.Length - pos - 1);

// Imprime os dados

Console.SetCursorPosition(5, lin);

Console.WriteLine(num);

Console.SetCursorPosition(16, lin);

Console.WriteLine(nome);

Console.SetCursorPosition(40, lin);

Console.WriteLine(turma);

lin++;

}

consultar.Close(); // fecha o ficheiro

}

else

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("\* MENSAGEM \* » Não existe nenhum ficheiro de registo.");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

Console.ReadLine();

}

**//----------------------------------**

**// 3 - CONSULTA ALUNOS POR TURMA**

**//----------------------------------**

// Nesta opção, é pedida qual a turma que deseja consultar,

// e então é listada todos os alunos desta mesma turma

static void consultaAlunosTurma()

{

Console.Clear();

int lin = 5;

string ficheiroRegisto = @"alunos\_encript.txt";

StreamReader consultar;

bool existeRegisto = false;

// verifica se o ficheiro existe

if (File.Exists(ficheiroRegisto) == true)

{

// Pede a turma que o utilizador deseja listar

Console.Write("Turma? ");

string turmaProcurada = Console.ReadLine();

Console.SetCursorPosition(5, lin - 3); Console.WriteLine("-----------------------------------------");

Console.SetCursorPosition(5, lin - 2); Console.WriteLine("NUMERO\tNOME\t\t\tTURMA");

Console.SetCursorPosition(5, lin - 1); Console.WriteLine("-----------------------------------------");

consultar = File.OpenText(ficheiroRegisto); // abre o ficheiro para leitura

string linha = "";

while ((linha = consultar.ReadLine()) != null) // Lê ficheiro até ao final

{

linha = desencriptar(linha); // DESENCRIPTAR LINHA DO REGISTO

int pos = linha.IndexOf(";");

if (linha.Substring

(linha.LastIndexOf(";") + 1, (linha.Length - 1) - (linha.LastIndexOf(";")))

== turmaProcurada)

// if (linha[linha.Length - 1] == turmaProcurada) // outra opcao para caso de turma com uma letra

{

string num = linha.Substring(0, pos); //Obtem o 1º campo de cada linha

// Refaz a variavel linha. a partir do 1º ";" até ao final da linha

linha = linha.Substring(pos + 1, linha.Length - pos - 1);

pos = linha.IndexOf(";");

string nome = linha.Substring(0, pos);

string turma = linha.Substring(pos + 1, linha.Length - pos - 1);

// Imprime os dados

Console.SetCursorPosition(5, lin);

Console.WriteLine(num);

Console.SetCursorPosition(16, lin);

Console.WriteLine(nome);

Console.SetCursorPosition(40, lin);

Console.WriteLine(turma);

lin++;

existeRegisto = true;

}

}

consultar.Close(); // fecha o ficheiro

// Mensagem caso não tenha nenhum registo da turma pesquisada

if (existeRegisto == false)

{

Console.SetCursorPosition(5, lin); Console.WriteLine("Nao há registos desta turma \"{0}\"", turmaProcurada);

}

}

else

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("\* MENSAGEM \* » Não existe nenhum ficheiro de registo.");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

Console.ReadLine();

}

**//----------------------------------**

**// 4 - ALTERAR REGISTO**

**//----------------------------------**

static void alterarRegisto()

{

string ficheiroRegisto = @"alunos\_encript.txt";

string ficheiroRegisto\_temp = @"alunos\_encript\_temp.txt";

Console.Write("\nInsira o número do aluno que deseja alterar do registo: ");

string registoAluno = Console.ReadLine();

StreamReader sr;

StreamWriter sw;

sr = File.OpenText(ficheiroRegisto);

sw = File.CreateText(ficheiroRegisto\_temp);

string linha = "";

bool encontrou = false;

while ((linha = sr.ReadLine()) != null)

{

// DESENCRIPTAR LINHA DO REGISTO

linha = desencriptar(linha);

// CRIA UM ARRAY COM 3 CAMPOS

string[] campos = new string[3];

// COLOCA CADA PARTE DA LINHA DENTRO DE UM CAMPO DO ARRAY

// CADA UMA DESTAS PARTES DIVIDIDAS PELO ";"

campos = linha.Split(';');

if (registoAluno == campos[0])

{

encontrou = true;

Console.WriteLine("\nAlteração para o usuário: \nNº: {0}\nNome: {1}\nTurma: {2}", campos[0], campos[1], campos[2]);

Console.Write("\nNovo nº: "); campos[0] = Console.ReadLine();

Console.Write("Novo nome: "); campos[1] = Console.ReadLine();

Console.Write("Nova turma: "); campos[2] = Console.ReadLine();

// Concatena informação do novo user

linha = campos[0] + ";" + campos[1] + ";" + campos[2];

// Encripta a linha nova

linha = encriptar(linha);

sw.WriteLine(linha);

}

else

{

linha = encriptar(linha);

sw.WriteLine(linha);

}

}

sr.Close();

sw.Close();

File.Delete(ficheiroRegisto);

File.Move(ficheiroRegisto\_temp, @"alunos\_encript.txt");

if (encontrou == false)

{

Console.WriteLine("Este aluno com o número {0} não existe.", registoAluno);

}

Console.ReadLine();

}

**//----------------------------------**

**// 5 - DELETAR REGISTO**

**//----------------------------------**

static void deletarRegisto()

{

consultaGeralAlunos();

string ficheiroRegisto = @"alunos\_encript.txt";

string ficheiroRegisto\_temp = @"alunos\_encript\_temp.txt";

Console.Write("Insira o número do aluno que deseja remover do registo: ");

string registoAluno = Console.ReadLine();

bool encontrou = true;

StreamReader sr;

StreamWriter sw;

sr = File.OpenText(ficheiroRegisto);

sw = File.CreateText(ficheiroRegisto\_temp);

string linha = "";

while ((linha = sr.ReadLine()) != null)

{

// DESENCRIPTAR LINHA DO REGISTO

linha = desencriptar(linha);

if (registoAluno != linha.Substring(0, linha.IndexOf(";")))

{

linha = encriptar(linha);

sw.WriteLine(linha);

}

else

{

Console.Write("Foi deletado o aluno com o registo nº {0} ", registoAluno);

encontrou = false;

}

}

sr.Close();

sw.Close();

File.Delete(ficheiroRegisto);

File.Move(ficheiroRegisto\_temp, @"alunos\_encript.txt");

if (encontrou == true)

{

Console.Write("\nAluno com o registo nº {0} não encontrado.", registoAluno);

}

Console.ReadLine();

}

**//----------------------------------**

**// Programa Principal**

**//----------------------------------**

static void Main(string[] args)

{

char opcao = ' ';

do

{

Console.Clear();

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

Console.WriteLine("\tMENU");

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

Console.WriteLine("1 - Criar Registo de Aluno");

Console.WriteLine("2 - Consultar lista geral de alunos");

Console.WriteLine("3 - Consultar alunos por turma");

Console.WriteLine("\n - ----- Opções Bônus -----");

Console.WriteLine("4 - Alterar Registo de Alulno");

Console.WriteLine("5 - Deletar Um Registo");

Console.WriteLine("0 - Sair");

Console.Write("\nOpção: ");

opcao = Convert.ToChar(Console.ReadLine());

switch (opcao)

{

case '1':

registoAluno();

break;

case '2':

consultaGeralAlunos();

break;

case '3':

consultaAlunosTurma();

break;

case '4':

alterarRegisto();

break;

case '5':

deletarRegisto();

break;

default:

if (opcao != '0')

Console.WriteLine("Opção inválida!");

else

Console.WriteLine("Saindo do programa... <\"ENTER\" para sair>");

break;

}

} while (opcao != '0');

}

}

}

**Ficha 08, questão 03 | VERSAO SIMPLIFICADA** => Programa que dado um username, gera uma palavra-chave e mostra na tela

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

Console.Write("Intro duza um username: ");

string user = Console.ReadLine();

string pass = "";

Random rand = new Random();

for (int i = 0; i < user.Length; i++)

{

if ((i + 1) % 2 == 0) // ou seja, se a posição da letra é par

{

int num = rand.Next(1,10);

pass = pass + Convert.ToString(num);

}

else

{

pass = pass + user[i];

}

}

Console.WriteLine("\nusername: {0}\npalavra-passe: {1}", user, pass);

// Console.ReadLine();

}

}

}

**Ficha 08, questão 03 | Completo** => Programa que dado um unername, gera uma palavra-chave, encripa-o e guarda em ficheiro

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

string user= " "; // propositadamente o user tem um espaço para que entre dentro do ciclo while

while (user.IndexOf(" ") != -1) // enquanto user tiver um espaço (quando encontrar um espaço vai retornar um inteiro positivo,

// logo, quando não tiver um espaço, vai ser então diferente de -1 => "i != -1")

{

Console.Write("Intro duza um username: ");

user = Console.ReadLine();

if (user.IndexOf(" ") != -1) // -1 significa que não foi encontrado o caractere pesquisado

{

Console.Write("\nUsername inválido!!!\n\n");

}

else

{

string pass = "";

Random rand = new Random(); // Tem que ficar fora do ciclo for. Porque ele vai sempre buscar o mesmo objeto. Só pode ser declarado uma vez

for (int i = 0; i < user.Length; i++)

{

if ((i + 1) % 2 == 0) // ou seja, se a posição da letra é par

{

int num = rand.Next(1, 10);

pass = pass + Convert.ToString(num);

}

else

{

pass = pass + user[i];

}

}

// Cria ou abre o ficheiro

string utilizadores = @"utilizadores.txt";

StreamWriter sw;

if (File.Exists(utilizadores) == true)

{

sw = File.AppendText(utilizadores);

}

else

{

sw = File.CreateText(utilizadores);

}

// Grava no ficheiro

pass = encriptar(pass);

string linha = user + ";" + pass;

sw.WriteLine(linha);

sw.Close();

// Mensagem final

Console.WriteLine("\nusername: {0}\npalavra-passe: {1}", user, pass);

}

} // while

Console.ReadLine();

}

static string encriptar(string pass)

{

string newPass = "";

for (int i = 0; i < pass.Length; i++)

{

int code = Convert.ToInt16(pass[i]);

code = code + 2;

newPass = newPass + Convert.ToChar(code);

}

return newPass;

}

}

}

**Ficha 08 | Cifra de Cesar com codigo ASCII**

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

string texto = " ";

int chave = 0;

Console.Write("Insira um texto: ");

texto = Console.ReadLine();

Console.Write("Insira um valor para a chave: ");

chave = Convert.ToInt16(Console.ReadLine());

// Obter comprimento do texto inserido

int comp = texto.Length;

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

{

int code = Convert.ToInt16(texto[i]);

Console.WriteLine("Código ASII convertido + chave: " + (code + chave));

Console.WriteLine("Novo caracter: " + Convert.ToChar(code + chave));

} // fim do for

Console.ReadLine();

} } }

Ficha 08 | Cifra de Cesar simplificado somente com o alfabeto

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

// string[] abc = new string[26] { "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q","r", "s", "t", "u", "v", "w", "x", "y", "z" };

// char[] abc = new char[26] { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z' };

/\* se quiser fazer com array, para acessar faz-se:

int pos1 = Array.IndexOf(teste, "d");

Console.WriteLine("pos1 = " + pos1); \*/

string texto = " ";

int chave = 0;

char aux = ' ';

string abc = "abcdefghijklmnopqrstuvwxyz";

Console.Write("Insira um texto: ");

texto = Console.ReadLine();

Console.Write("Insira um valor para a chave: ");

chave = Convert.ToInt16(Console.ReadLine());

// Obter comprimento do texto inserido

int comp = texto.Length;

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

{

aux = texto[i];

int pos = abc.IndexOf(aux);

if (pos + chave > 25) // resolver para o caso menor que zero: || pos + chave < 0)

Console.Write(abc[(pos + chave) % 26]);

else if (pos + chave < 0)

{

while (pos + chave < 0)

{

chave = chave + 26;

}

Console.Write(abc[pos + chave]);

}

else

Console.Write(abc[pos + chave]);

} // fim do for

Console.ReadLine();

}

}

}

### Ficha 09 | DateTime

**// Ficha 09 | Exercicio 01** - Registar temperatura

// Exercícios sobre DateTime

// obter data

// obter hora

// gerar temperatura

// obter designacao do ficheiro

// guardar ficheiro

// <- Tudo dentro de um ciclo

**using System.Threading;**

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

Console.Clear();

Console.WriteLine("\nSIMULADOR DE REGISTO DE TEMPERATURA\nFICHA 09 | EXERCICIO 01\n");

Console.SetCursorPosition(10, 3);

Console.WriteLine("Data");

Console.SetCursorPosition(25, 3);

Console.WriteLine("Hora");

Console.SetCursorPosition(40, 3);

Console.WriteLine("Temperatura");

Console.SetCursorPosition(10, 4);

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

Random rand = new Random();

int lin = 5;

int op = 1;

while (op == 1)

{

string hoje = DateTime.Today.ToString("yyyy-MM-dd");

Console.SetCursorPosition(10, lin);

Console.WriteLine(hoje);

string hora = DateTime.Now.ToString("HH:mm:ss");

Console.SetCursorPosition(25, lin);

Console.WriteLine(hora);

int temperatura = rand.Next(10, 26);

Console.SetCursorPosition(40, lin);

Console.WriteLine(temperatura.ToString());

// FICHEIRO

string diaDoAno = DateTime.Today.DayOfYear.ToString("###"); // cada cardinal significa um algarismo

string ficheiro = @"temp" + diaDoAno + ".txt";

StreamWriter sw;

if (File.Exists(ficheiro) == true)

{

sw = File.AppendText(ficheiro);

}

else

{

sw = File.CreateText(ficheiro);

}

string linha = hoje + ";" + hora + ";" + temperatura;

sw.WriteLine(linha);

sw.Close();

Thread.Sleep(3000);

lin++;

// Para evitar o erro da gama de valores da linha

if (lin == 300)

{

Console.Clear();

lin = 5;

}

}

//Console.ReadLine();

}

}

}

**Ficha 09 - DateTime | Exercicio 2** - Consulta dos dados do simulador do exercício 01

// ler data

// obter nº dia do ano

// calcula min, máx e média das temperaturas

**using System.Threading;**

namespace Exercicios

{

class Program

{

static void Main(string[] args)

{

Console.Write("------------------------------------------------------------------------\n");

Console.Write("- Ficha 9 - Questão 2 | Consulta dos dados do simulador do exercício 1 -\n");

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

Console.Write("Insira uma data no formato \"yyyy-mm-dd\": ");

DateTime dataUser = Convert.ToDateTime(Console.ReadLine());

string diaDoAno = Convert.ToString(dataUser.DayOfYear); // obtem o dia do ano em funcao da data que o user introduziu

string ficheiro = @"temp" + diaDoAno + ".txt"; // declara e assina uma variável string com o nome tempXXX.txt, onde XXX é o dia do ano

StreamReader sr; // Inicia o Stream para leitura de ficheiros (StreamReader)

if (File.Exists(ficheiro) == true)

{

Console.WriteLine("\nData\t\tHora\t\tTemperatura");

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

sr = File.OpenText(ficheiro); // abre o ficheiro para leitura

string linha = "";

int max = int.MinValue;

int min = int.MaxValue;

double media = 0;

int somatorio = 0;

int cont = 0;

while ((linha = sr.ReadLine()) != null) // lê o ficheiro até o final

{

string[] campos = new string[3];

campos = linha.Split(';'); // a função Split separa "linha" pelos ";" para um array

if (campos[0] == Convert.ToString(dataUser.ToString("yyyy-MM-dd")))

{

Console.WriteLine(campos[0] + "\t" + campos[1] + "\t" + campos[2]);

int temp = Convert.ToInt16(campos[2]);

if (temp > max)

{

max = temp;

}

if (temp < min)

{

min = temp;

}

somatorio = somatorio + temp;

cont++;

}

}

// Calculo da média

media = somatorio \* 1.0 / cont; // "\* 1.0" serve para transformar o valor para real.

Console.WriteLine("\nValor temperatura mínima: {0}\nValor temperatura máxima: {1}\nMédia da temperatura: {2}", min, max, media);

}

else

{

Console.Write("O ficheiro não existe.");

}

Console.ReadLine();

}

}

}