

EJERCICIOS TEMA 4 “ARRAYS Y ESTRUCTURAS”

(4.1.1.1) Un programa que pida al usuario 4 números, los memorice (utilizando una tabla), calcule su media aritmética y después muestre en pantalla la media y los datos tecleados.

```
using System;
```

```
public class ejercicio
```

```
{  
    public static void Main ()  
    {  
        float [] ejemplo = new float[4];  
        float media;  
  
        for (int i = 0; i <= 3; i++) {  
            Console.Write ("Introduzca número:", i);  
            ejemplo [i] = Convert.ToSingle (Console.ReadLine ());  
        }  
  
        media = (ejemplo[0]+ejemplo[1]+ ejemplo[2]+ ejemplo[3])/4;  
  
        Console.Write ("La media de {0}, {1}, {2} y {3} es: {4}", ejemplo[0], ejemplo [1],  
ejemplo [2], ejemplo [3], media);  
    }  
}
```

```
using System;
```

```
public class ejercicio
```

```
{  
    public static void Main ()  
    {  
        float [] ejemplo = new float[4];  
        float media;  
        int cont = 1;  
        int i = 0;  
  
        do {  
            Console.Write ("Introduzca el {0}º número: ", cont);  
            ejemplo [i] = Convert.ToSingle (Console.ReadLine ());  
            cont++;  
            i++;  
        }  
    }  
}
```

```

    } while (cont < 5);

    media = (ejemplo[0]+ejemplo[1]+ ejemplo[2]+ ejemplo[3])/4;

    Console.WriteLine("\nLa media de {0}, {1}, {2} y {3} es: {4}", ejemplo[0], ejemplo [1],
ejemplo [2], ejemplo [3], media);
    }
}

```

(4.1.1.2) Un programa que pida al usuario 5 números reales (pista: necesitarás un array de "float") y luego los muestre en el orden contrario al que se introdujeron.

```
using System;
```

```

public class ejercicio
{
    public static void Main ()
    {
        float [] ejemplo = new float[5];
        int cont = 1;
        int i = 0;

        do {
            Console.WriteLine("Introduzca el {0}º número: ", cont);
            ejemplo [i] = Convert.ToSingle (Console.ReadLine ());
            cont++;
            i++;
        } while (cont < 6);

        Console.WriteLine("\n{0}, {1}, {2}, {3} y {4}", ejemplo[4], ejemplo [3], ejemplo [2], eje
mplo [1], ejemplo[0]);
    }
}

```

(4.1.2.1) Un programa que almacene en una tabla el número de días que tiene cada mes (supondremos que es un año no bisiesto), pida al usuario que le indique un mes (1=enero, 12=diciembre) y muestre en pantalla el número de días que tiene ese mes.

```
using System;
```

```
public class ejercicio
```

```
{  
    public static void Main ()  
    {  
        int[] mes = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  
        int num;  
  
        Console.WriteLine ("Introduzca el número del mes (1-12): ");  
        num = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.WriteLine ("El mes que ha elegido tiene {0} días.", mes [num-1]);  
        // Pongo num-1 porque el array se lee de 0 a 11.  
    }  
}
```

(4.1.3.3) Un programa que pida al usuario 10 números, calcule su media y luego muestre los que están por encima de la media.

```
using System;
```

```
public class ejercicio
```

```
{
    public static void Main ()
    {
        float[] ejemplo= new float[10];
        int i = 0;
        float media, suma=0;

        do {
            Console.Write ("Introduce el {0}º número: ", i + 1);
            ejemplo [i] = Convert.ToSingle (Console.ReadLine ());
            i++;
        } while (i < 10);

        Console.WriteLine ();
        Console.Write ("Los números introducidos por el usuario son: ");

        for (i = 0; i <= 9; i++) {
            suma += ejemplo [i];
            Console.Write (" {0}, ", ejemplo [i]);
        }

        Console.WriteLine ();
        media = suma/10;
        Console.WriteLine ("\nLa media de los números introducidos es {0}", media);

        Console.WriteLine ();

        for (i = 0; i <= 9; i++) {
            if (ejemplo [i] >= media) {
                Console.Write ("El número {0} está por encima de la media", ejemplo [i]);
                Console.WriteLine ();
            }
        }
    }
}
```

(4.1.3.4) Un programa que almacene en una tabla el número de días que tiene cada mes (de un año no bisiesto), pida al usuario que le indique un mes (ej. 2 para febrero) y un día (ej. el día 15) y diga qué número de día es dentro del año (por ejemplo, el 15 de febrero sería_ el día número 46, el 31 de diciembre sería el día 365).

```
using System;
```

```
public class ejemplo
```

```
{
    public static void Main()
    {
        int[] meses= {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        int mes, dia, suma=0, total;

        Console.Write ("Introduce el mes (1/12): ");
        mes = Convert.ToInt32 (Console.ReadLine ());
        Console.Write ("Introduce el día: ");
        dia = Convert.ToInt32 (Console.ReadLine ());

        for (int i = 0; i < mes - 1; i++) {
            suma += meses [i];
        }
        total = suma + dia;

        Console.Write ("El número del día dentro del año es: {0}", total);
    }
}
```

```
using System;
```

```
public class ejemplo
```

```
{
    public static void Main()
    {
        int[] meses= {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        int mes, dia, suma=0, total;

        try{

            Console.Write ("Introduce el mes (1/12): ");
            mes = Convert.ToInt32 (Console.ReadLine ());
            Console.Write ("Introduce el día: ");
            dia = Convert.ToInt32 (Console.ReadLine ());

            if (dia>31){
```

```
        throw new Exception ("Un mes no tiene tantos días.");
    }

    for (int i = 0; i < mes - 1; i++) {
        suma += meses [i];
    }
    total = suma + día;

    Console.WriteLine ("El número del día dentro del año es: {0}", total);
}
catch (IndexOutOfRangeException index){
    Console.WriteLine ("Error de índice.", index.Message);

}
catch (Exception ex){
    Console.WriteLine ("Ha habido un error: "+ ex.Message);
}
}
```

Crear un programa que pida al usuario 10 números en coma flotante (pista: necesitarás un array de "float") y luego los muestre en orden inverso (del último que se ha introducido al primero que se introdujo).

```
using System;
```

```
public class ejercicio
```

```
{  
    public static void Main ()  
    {  
        float[] ejemplo = new float[10];  
        int i=0, j=9;  
  
        do  
        {  
            Console.Write ("Introduce el {0}º número: ", i+1);  
            ejemplo [i] = Convert.ToSingle (Console.ReadLine ());  
            i++;  
        }  
        while (i<10);  
  
        //for (int j = 9; j >= 0; j--) {  
        while (j>=1){  
            Console.Write (" {0}, ", ejemplo [j]);  
            j--;  
        }  
        if (j == 0) {  
            Console.Write (ejemplo [0]);  
        }  
    }  
}
```

(4.1.3.6) Un programa que pida 10 nombres y los memorice (pista: esta vez se trata de un array de "string"). Después deberá pedir que se teclee un nombre y dirá si se encuentra o no entre los 10 que se han tecleado antes. Volverá a pedir otro nombre y a decir si se encuentra entre ellos, y así sucesivamente hasta que se teclee "fin".

```
using System;
```

```
public class ejercicio
```

```
{  
    public static void Main ()  
    {  
        string[] ejemplo = new string[10];  
        int i = 0;  
        string nombre;  
  
        do {  
            Console.Write ("Introduce el {0}º nombre para la base de datos: ", i + 1);  
            ejemplo [i] = Console.ReadLine ();  
            i++;  
        } while (i < 10);  
  
        do {  
            bool found=false;  
  
            Console.Write ("Introduzca un nombre: ");  
            nombre = Console.ReadLine ();  
  
            if (nombre=="fin"){  
                break;  
            }  
            for (i = 0; i < ejemplo.Length; i++) {  
                if (nombre == ejemplo [i]) {  
                    found=true;  
                    break;  
                }  
            }  
            if (found==true){  
                Console.WriteLine ("El nombre introducido está en el listado.");  
            }  
            else{  
                Console.WriteLine ("El nombre introducido NO está en el listado.");  
            }  
        } while (nombre != "fin");  
    }  
}
```


(4.1.3.7) Un programa que prepare espacio para un máximo de 100 nombres. El usuario deberá ir introduciendo un nombre cada vez, hasta que se pulse Intro sin teclear nada, momento en el que dejarán de pedirse más nombres y se mostrará en pantalla la lista de los nombres que se han introducido.

```
using System;
```

```
public class ejemplo
```

```
{  
    public static void Main ()  
    {  
        string[] ejemplo = new string[100];  
        int i = 0;  
  
        for (i = 0; i < ejemplo.Length; i++) {  
            Console.Write ("Introduce un nombre: ");  
            ejemplo[i] = Console.ReadLine ();  
            if (ejemplo [i] == "") {  
                break;  
            }  
        }  
        for (i = 0; i < ejemplo.Length; i++) {  
            if ((ejemplo [i] == null) || (ejemplo [i]=="")){  
                break;  
            }  
            Console.WriteLine ("- {0}",ejemplo [i]);  
        }  
    }  
}
```

(4.1.3.8) Un programa que reserve espacio para un vector de 3 componentes, pida al usuario valores para dichas componentes (por ejemplo [2, -5, 7]) y muestre su módulo (raíz cuadrada de la suma de sus componentes al cuadrado).

```
using System;
```

```
public class ejemplo
```

```
{  
    public static void Main ()  
    {  
        double[] ejemplo= new double[3];  
        double suma, modulo;  
        int i;  
  
        for (i = 0; i < ejemplo.Length; i++) {  
            Console.Write ("Introduce el {0}º número: ",i+1);  
            ejemplo [i] = Convert.ToDouble (Console.ReadLine ());  
        }  
  
        suma = 0;  
        for (i = 0; i < ejemplo.Length; i++) {  
            ejemplo [i] = Math.Pow (ejemplo [i], 2);  
            suma += ejemplo [i];  
        }  
  
        modulo = Math.Sqrt (suma);  
  
        Console.Write ("\nEl módulo de los números introducidos es: {0}\n", modulo.ToString("N2"));  
    }  
}
```

(4.1.3.9) Un programa que reserve espacio para dos vectores de 3 componentes, pida al usuario sus valores y calcule la suma de ambos vectores (su primera componente será x_1+y_1 , la segunda será x_2+y_2 y así sucesivamente).

```
using System;
```

```
public class ejemplo
```

```
{  
    public static void Main ()  
    {  
        double[] array1= new double[3];  
        double[] array2= new double[3];  
        double suma;  
        int i;  
  
        for (i = 0; i < array1.Length; i++) {  
            Console.Write ("Introduce el {0}º número del primer array: ",i+1);  
            array1 [i] = Convert.ToDouble (Console.ReadLine ());  
        }  
        Console.WriteLine ();  
        for (i = 0; i < array2.Length; i++) {  
            Console.Write ("Introduce el {0}º número del segundo array: ",i+1);  
            array2 [i] = Convert.ToDouble (Console.ReadLine ());  
        }  
        Console.WriteLine ();  
        for (i = 0; i < array2.Length; i++) {  
            suma = array1 [i] + array2 [i];  
            Console.WriteLine ("La suma de la {0}º posición de los vectores es: {1}", i+1, suma);  
        }  
    }  
}
```

(4.1.3.10) Un programa que reserve espacio para dos vectores de 3 componentes, pida al usuario sus valores y calcule su producto escalar ($x_1 \cdot y_1 + x_2 \cdot y_2 + x_3 \cdot y_3$).

```
using System;
```

```
public class ejemplo
```

```
{  
    public static void Main ()  
    {  
        double[] array1= new double[3];  
        double[] array2= new double[3];  
        double productoescalar, producto;  
        int i;  
  
        for (i = 0; i < array1.Length; i++) {  
            Console.Write ("Introduce el {0}º número del primer array: ",i+1);  
            array1 [i] = Convert.ToDouble (Console.ReadLine ());  
        }  
        Console.WriteLine ();  
        for (i = 0; i < array2.Length; i++) {  
            Console.Write ("Introduce el {0}º número del segundo array: ",i+1);  
            array2 [i] = Convert.ToDouble (Console.ReadLine ());  
        }  
        Console.WriteLine ();  
        productoescalar = 0;  
        for (i = 0; i < array2.Length; i++) {  
            producto = array1 [i] * array2 [i];  
            productoescalar += producto;  
        }  
        Console.WriteLine ("El producto escalar ((x1*y1)+(x2*y2)+(x3*y3))  
es: {0}", productoescalar);  
    }  
}
```

(4.1.3.11) Un programa que pida al usuario 4 números enteros y calcule (y muestre) cuál es el mayor de ellos. Nota: para calcular el mayor valor de un array, hay que comparar cada uno de los valores que tiene almacenados con el que hasta ese momento es el máximo. El valor inicial de este máximo no debería ser cero (porque fallaría si todos los números son negativos), sino el primer elemento del array.

```
using System;
```

```
public class ejemplo
```

```
{  
    public static void Main ()  
    {  
        int[] ejemplo= new int[4];  
        int i = 0, mayor;  
  
        do {  
            Console.Write ("Introduce el {0}º número: ", i + 1);  
            ejemplo [i] = Convert.ToInt32 (Console.ReadLine ());  
            i++;  
        } while (i < 4);  
  
        mayor = ejemplo [0];  
  
        for (i = 1; i < ejemplo.Length; i++) {  
            if (mayor < ejemplo [i]) {  
                mayor = ejemplo [i];  
            }  
        }  
        Console.Write ("El mayor de los números introducidos es {0}", mayor);  
    }  
}
```

(4.1.4.5) Un programa que prepare espacio para un máximo de 10 nombres. Deberá mostrar al usuario un menú que le permita realizar las siguientes operaciones:

- Añadir un dato al final de los ya existentes.
- Insertar un dato en una cierta posición (como ya se ha comentado, los que queden detrás deberán desplazarse "a la derecha" para dejarle hueco; por ejemplo, si el array contiene "hola", "adios" y se pide insertar "bien" en la segunda posición, el array pasará a contener "hola", "bien", "adios".
- Borrar el dato que hay en una cierta posición (como se ha visto, lo que estaban detrás deberán desplazarse "a la izquierda" para que no haya huecos; por ejemplo, si el array contiene "hola", "bien", "adios" y se pide borrar el dato de la segunda posición, el array pasará a contener "hola", "adios".
- Mostrar los datos que contiene el array.
- Salir del programa.

```
using System;
```

```
public class ejemplo
```

```
{
```

```
    public static void Main ()
```

```
    {
```

```
        string[] array= new string[10];
```

```
        int num, i, posicion, borrar;
```

```
        string palabra, palabra2;
```

```
        try{
```

```
            do{
```

```
                Console.Write ("\tMENÚ DE PROGRAMA\n\n1) Añadir un dato al final del array.\n2) Insertar un dato.\n3) Borrar un dato.\n4) Mostrar los datos del array.\n5) Salir.");
```

```
                Console.Write ("\n¿Qué desea hacer?: ");
```

```
                num = Convert.ToInt32 (Console.ReadLine ());
```

```
                if ((num<1)||(num>5)){
```

```
                    Console.Write ("El menú no tiene tantas opciones.\n\n");
```

```
                }
```

```
                if (num == 1) {
```

```
                    Console.Write ("¿Qué palabra desea añadir?: ");
```

```
                    palabra = Console.ReadLine ();
```

```

        array [9] = palabra;
        Console.Write ("\nHECHO!!\n\n");
    }
    if (num==2){
        Console.WriteLine ("¿Qué dato desea insertar?: ");
        palabra2= Console.ReadLine();
        Console.WriteLine ("¿En qué posición quiere insertarlo?: ");
        posicion=Convert.ToInt32 (Console.ReadLine());

        for (i=9; i>posicion; i--){
            array[i]=array [i-1];
        }
        array[posicion]=palabra2;

        Console.Write ("\nHECHO!!\n\n");
    }
    if (num==3){
        Console.Write ("¿Qué posición desea borrar?: ");
        borrar=Convert.ToInt32 (Console.ReadLine());
        for (i=borrar; i<=8; i++){
            array[i]=array[i+1];
        }
        Console.Write ("\nHECHO!!\n\n");
    }
    if (num==4){
        for (i=0; i<array.Length; i++){
            if (array[i]==null){
                array[i]="Posición vacía";
            }
            Console.Write ("- {0} ", array[i]);
            Console.WriteLine();
        }
        Console.Write ("\nHECHO!!\n\n");
    }
}
while (num!=5);

if (num == 5) {
    Console.Write ("El programa ha finalizado.");
}
}
catch (Exception error){
    Console.Write ("Ha habido un error: "+ error.Message);
}
}
}

```

// Haciéndolo con métodos:

```
using System;
```

```
public class ejemplo
```

```
{
```

```
    public static void Main ()
```

```
    {
```

```
        string[] array= new string[10];
```

```
        int num;
```

```
        try{
```

```
            do{
```

```
                Console.Write ("\tMENÚ DE PROGRAMA\n\n1) Añadir un dato al final del array.\n2) Insertar un dato.\n3) Borrar un dato.\n4) Mostrar los datos del array.\n5) Salir.");
```

```
                Console.Write ("\n¿Qué desea hacer?: ");
```

```
                num = Convert.ToInt32 (Console.ReadLine ());
```

```
                if ((num<1)||(num>5)){
```

```
                    Console.Write ("El menú no tiene tantas opciones.\n\n");
```

```
                }
```

```
                else{
```

```
                    procesarAccion(num, array);
```

```
                }
```

```
            }
```

```
            while (num!=5);
```

```
            if (num == 5) {
```

```
                Console.Write ("El programa ha finalizado.");
```

```
            }
```

```
        }
```

```
        catch (Exception error){
```

```
            Console.Write ("Ha habido un error: "+ error.Message);
```

```
        }
```

```
    }
```

```
    public static void procesarAccion(int accion,string[] array){
```

```
        switch (accion) {
```

```
            case 1:
```

```
                añadir (array);
```

```
                break;
```

```
            case 2:
```

```
                insertar (array);
```

```
                break;
```



```

        case 3:
            borrar (array);
            break;
        case 4:
            mostrar (array);
            break;
    }

}

public static void añadir(string[] array)
{
    string palabra;
    Console.Write ("¿Qué palabra desea añadir?: ");
    palabra = Console.ReadLine ();
    array [9] = palabra;
    Console.Write ("\nHECHO!!\n\n");
}

public static void insertar (string []array)
{
    string palabra2;
    int posicion;
    Console.WriteLine ("¿Qué dato desea insertar?: ");
    palabra2= Console.ReadLine();
    Console.WriteLine ("¿En qué posición quiere insertarlo?: ");
    posicion=Convert.ToInt32 (Console.ReadLine());

    for (int i=9; i>posicion; i--){
        array[i]=array [i-1];
    }
    array[posicion]=palabra2;

    Console.Write ("\nHECHO!!\n\n");
}

public static void borrar(string [] array)
{
    Console.Write ("¿Qué posición desea borrar?: ");
    int borrar=Convert.ToInt32 (Console.ReadLine());
    for (int i=borrar; i<=8; i++){
        array[i]=array[i+1];
    }
    Console.Write ("\nHECHO!!\n\n");
}

public static void mostrar (string [] array)
{
    for (int i=0; i<array.Length; i++){
        if (array[i]==null){
            array[i]="Posición vacía";

```

```

    }
    Console.Write ("- {0} ", array[i]);
    Console.WriteLine();
}
Console.Write ("\nHECHO!!\n\n");
}
}

```

(4.2.1) Un programa que pida al usuario dos bloques de 10 números enteros (usando un array de dos dimensiones). Después deberá mostrar el mayor dato que se ha introducido en cada uno de ellos.

```

using System;
public class ejemplo
{
    public static void Main()
    {
        int[][] ejemplo;
        ejemplo=new int[2][];
        ejemplo[0]=new int [10];
        ejemplo[1]=new int[10];
        int mayor;

        for (int i = 0; i < ejemplo.Length; i++) {
            for (int j = 0; j < ejemplo [i].Length; j++) {
                Console.Write ("Introduzca el {0}º número del {1}º array: ", j + 1, i + 1);
                ejemplo [i] [j] = Convert.ToInt32 (Console.ReadLine ());
            }
        }
        for (int i=0; i<ejemplo.Length; i++){
            for (int j = 0; j < ejemplo [i].Length; j++) {
                Console.Write (" {0} ", ejemplo [i] [j]);
            }
            Console.WriteLine ();
        }
        mayor =ejemplo[0][0];
        for (int i = 0; i < ejemplo.Length; i++) {
            for (int j = 0; j < ejemplo [i].Length; j++) {
                if (mayor<ejemplo[i][j]){
                    mayor = ejemplo [i] [j];
                }
            }
        }
        Console.Write ("El mayor número introducido en los arrays es: {0}", mayor);
    }
}

```

(4.2.2) Un programa que pida al usuario dos bloques de 6 cadenas de texto. Después pedirá al usuario una nueva cadena y comprobará si aparece en alguno de los dos bloques de información anteriores.

```
using System;
```

```
public class ejemplo
```

```
{  
    public static void Main()  
    {  
        string[][] cadena;  
        cadena=new string[2][];  
        cadena [0] = new string[6];  
        cadena [1] = new string[6];  
        string palabra;  
        bool existe=false;  
  
        for (int i = 0; i < cadena.Length; i++) {  
            for (int j = 0; j < cadena [i].Length; j++) {  
                Console.Write ("Introduce la {0}ª palabra del {1}º array: ", j + 1, i + 1);  
                cadena[i][j]=Console.ReadLine();  
            }  
        }  
        Console.Write ("\n\nIntroduce una nueva palabra: ");  
        palabra = Console.ReadLine ();  
  
        for (int i = 0; i < cadena.Length; i++) {  
            for (int j = 0; j < cadena [i].Length; j++) {  
                if (palabra == cadena [i] [j]) {  
                    existe = true;  
                }  
            }  
        }  
        if (existe == true) {  
            Console.Write ("\n\nLa palabra introducida ya se encuentra en los arrays.");  
        }  
        else {  
            Console.Write ("\n\nLa palabra introducida NO se encuentra en los arrays.");  
        }  
    }  
}
```

(4.3.1.1) Un "struct" que almacene datos de una canción en formato MP3: Artista, Título, Duración (en segundos), Tamaño del fichero (en KB). Un programa debe pedir los datos de una canción al usuario, almacenarlos en dicho "struct" y después mostrarlos en pantalla.

```
using System;
```

```
public class ejemplo
```

```
{  
    struct cancion  
    {  
        public float duracion;  
        public string titulo;  
        public string artista;  
        public int tamaño;  
    }  
}
```

```
public static void Main ()
```

```
{  
    cancion mp3;  
  
    Console.Write ("Introduce el artista de la canción: ");  
    mp3.artista = Console.ReadLine ();  
    Console.Write ("Introduce el título de la canción: ");  
    mp3.titulo = Console.ReadLine ();  
    Console.Write ("Introduce la duración de la canción: ");  
    mp3.duracion = Convert.ToSingle (Console.ReadLine ());  
    Console.Write ("Introduce el tamaño de la canción: ");  
    mp3.tamaño = Convert.ToInt32 (Console.ReadLine ());  
}
```

```
    Console.WriteLine ("\nArtista: {0}\nTítulo: {1}\nDuración: {2} minutos\nTamaño: {3} KB", mp3.artista, mp3.titulo, mp3.duracion, mp3.tamaño);  
}
```

(4.3.2.1) Ampliar el programa 4.3.1.1, para que almacene datos de hasta 100 canciones. Deberá tener un menú que permita las opciones: añadir una nueva canción, mostrar el título de todas las canciones, buscar la canción que contenga un cierto texto (en el artista o en el título).

```
using System;
```

```
public class estructura
```

```
{  
    struct cancion  
    {  
        public string artista;  
        public string titulo;  
        public int duracion;  
        public float peso;  
    }  
    public static void Main ()  
    {  
        cancion[] mp3=new cancion[3];  
        int num, contador=0;  
        string buscador;
```

```
        inicio:
```

```
        try{
```

```
            do {
```

```
                Console.Write ("\n\tMENÚ DE PROGRAMA\n\n1) Añadir canción.\n2) Mostrar  
artistas y títulos.\n3) Buscar artista o título.\n4) Salir.");
```

```
                Console.Write ("\n\n¿Qué desea hacer?: ");
```

```
                num = Convert.ToInt32 (Console.ReadLine ());
```

```
                if (num == 1) {
```

```
                    if (contador == 3) {
```

```
                        Console.Write ("\nEl array está lleno.\n");
```

```
                        goto inicio;
```

```
                    }
```

```
                    Console.Write ("Introduzca el nombre del artista: ");
```

```
                    mp3 [contador].artista = Console.ReadLine ();
```

```
                    Console.Write ("Introduzca el título de la canción: ");
```

```
                    mp3 [contador].titulo = Console.ReadLine ();
```

```
                    Console.Write ("Introduzca la duración de la canción: ");
```

```
                    mp3 [contador].duracion = Convert.ToInt32 (Console.ReadLine ());
```

```
                    Console.Write ("Introduzca el peso de la canción en KB: ");
```

```
                    mp3 [contador].peso = Convert.ToSingle (Console.ReadLine ());
```

```
                    contador++;
```

```

    }
    if (num==2){
        for (int i=0; i<mp3.Length; i++){
            if (mp3[i].artista==null){
                break;
            }
            Console.WriteLine ("El artista es: {0} y el título es: {1}", mp3[i].artista, mp3[i].t
itulo);
            Console.WriteLine();}
        }
        if (num==3){
            bool encontrado=false;
            Console.WriteLine ("Introduzca el artista/título que quiere buscar: ");
            buscador=Console.ReadLine();
            for (int i=0; i<mp3.Length; i++){
                if ((buscador==mp3[i].artista)|| (buscador==mp3[i].titulo)){
                    encontrado=true;
                }
            }
            if (encontrado==true){
                Console.WriteLine ("El artista/título que buscaba está en la base de datos.\n");
            }
            else Console.WriteLine ("El artista/título que buscaba NO está en la base de datos
.\n");
        }
        if ((num<1)|| (num>4)){
            Console.WriteLine ("Elija una opción del menú entre 1 y 4.\n");
        }
    } while (num != 4);

    if (num == 4) {
        Console.WriteLine ("El programa ha finalizado.");
    }
}
catch (Exception error){
    Console.WriteLine ("Ha habido un error: {0}\n", error.Message);
    goto inicio;
}
}
}

```

(4.3.2.2) Un programa que permita guardar datos de "imágenes" (ficheros de ordenador que contengan fotografías o cualquier otro tipo de información gráfica). De cada imagen se debe guardar: nombre (texto), ancho en píxeles (por ejemplo 2000), alto en píxeles (por ejemplo, 3000), tamaño en Kb (por ejemplo 145,6). El programa debe ser capaz de almacenar hasta 700 imágenes (deberá avisar cuando su capacidad esté llena). Debe permitir las opciones: añadir una ficha nueva, ver todas las fichas (número y nombre de cada imagen), buscar la ficha que tenga un cierto nombre.

```
using System;
```

```
public class estructura2  
{
```

```
    struct imagen
```

```
    {  
        public string nombre;  
        public int ancho;  
        public int alto;  
        public float peso;  
    }
```

```
    public static void Main ()
```

```
    {  
        imagen[] jpg=new imagen[3];  
        int num, contador = 0;  
        string buscador;
```

```
        do{
```

```
            Console.Write ("\n\tMENÚ DE PROGRAMA\n\n1)Añadir una imagen.\n2)Mostrar  
datos.\n3)Buscar una imagen por nombre.\n4)Salir.\n¿Qué opción elige?: ");  
            num=Convert.ToInt32 (Console.ReadLine());
```

```
            switch (num){
```

```
            case 1:
```

```
                if (contador==3){
```

```
                    Console.Write ("\nEl array está lleno.\n");
```

```
                } else{
```

```
                    Console.Write ("Introduzca el nombre de la imagen: ");
```

```
                    jpg[contador].nombre=Console.ReadLine();
```

```
                    Console.Write ("Introduzca el alto de la imagen en píxeles: ");
```

```
                    jpg[contador].alto=Convert.ToInt32 (Console.ReadLine());
```

```
                    Console.Write ("Introduzca el ancho de la imagen en píxeles: ");
```

```
                    jpg[contador].ancho=Convert.ToInt32 (Console.ReadLine());
```

```
                    Console.Write ("Introduzca el tamaño de la imagen en Kb: ");
```

```
                    jpg[contador].peso=Convert.ToSingle (Console.ReadLine());
```

```
                    contador++;
```

```
                }
```

```
                break;
```

```

case 2:
    for (int i=0; i<jpg.Length; i++){
        if (jpg[i].nombre!=null){
            Console.Write ("\nNombre de la {0}° imagen: {1}\n", i+1, jpg[i].nombre);
        }
    }
    break;
case 3:
    bool encontrado=false;
    Console.Write ("Introduzca el texto que quiere buscar: ");
    buscador=Console.ReadLine();
    for (int i=0; i<jpg.Length; i++){
        if (buscador==jpg[i].nombre){
            encontrado=true;
            break;
        }
    }
    if (encontrado==true){
        Console.Write("\nEl nombre buscado está en la base de datos.\n");
    } else {
        Console.Write ("\nEl nombre buscado NO está en la base de datos.\n");
    }

    break;
}
}
while (num!=4);

if (num == 4) {
    Console.Write ("\nFin de programa.\n");
}
}
}

```


(4.3.3.1) Ampliar el programa 4.3.2.1, para que el campo "duración" se almacene como minutos y segundos, usando un "struct" anidado que contenga a su vez estos dos campos.

```
using System;
```

```
public class estructura
```

```
{
```

```
    struct duracion
```

```
    {
```

```
        public int minutos;
```

```
        public int segundos;
```

```
    }
```

```
    struct cancion
```

```
    {
```

```
        public string artista;
```

```
        public string titulo;
```

```
        public duracion tiempo;
```

```
        public float peso;
```

```
    }
```

```
    public static void Main ()
```

```
    {
```

```
        cancion[] mp3=new cancion[3];
```

```
        int num, contador=0;
```

```
        string buscador;
```

```
        inicio:
```

```
        try{
```

```
            do {
```

```
                Console.Write ("\tMENÚ DE PROGRAMA\n\n1) Añadir canción.\n2) Mostrar  
artistas y títulos.\n3) Buscar artista o título.\n4) Salir.");
```

```
                Console.Write ("\n¿Qué desea hacer?: ");
```

```
                num = Convert.ToInt32 (Console.ReadLine ());
```

```
                if (num == 1) {
```

```
                    if (contador == 3) {
```

```
                        Console.Write ("\nEl array está lleno.\n");
```

```
                        goto inicio;
```

```
                    }
```

```
                    Console.Write ("Introduzca el nombre del artista: ");
```

```
                    mp3 [contador].artista = Console.ReadLine ();
```

```
                    Console.Write ("Introduzca el título de la canción: ");
```

```
                    mp3 [contador].titulo = Console.ReadLine ();
```

```
                    Console.Write ("Introduzca la duración de la canción en minutos: ");
```

```

        mp3 [contador].tiempo.minutos = Convert.ToInt32 (Console.ReadLine ());
        Console.WriteLine ("Introduzca la duración de la canción en segundos: ");
        mp3 [contador].tiempo.segundos = Convert.ToInt32 (Console.ReadLine ());
        Console.WriteLine ("Introduzca el peso de la canción en KB: ");
        mp3 [contador].peso = Convert.ToSingle (Console.ReadLine ());
        contador++;
    }
    if (num==2){
        for (int i=0; i<mp3.Length; i++){
            if (mp3[i].artista==null){
                break;
            }
            Console.WriteLine ("El artista es: {0} y el título es: {1}", mp3[i].artista, mp3[i].t
itulo);
            Console.WriteLine();
        }
        if (num==3){
            bool encontrado=false;
            Console.WriteLine ("Introduzca el artista/título que quiere buscar: ");
            buscador=Console.ReadLine();
            for (int i=0; i<mp3.Length; i++){
                if ((buscador==mp3[i].artista)||(buscador==mp3[i].titulo)){
                    encontrado=true;
                }
            }
            if (encontrado==true){
                Console.WriteLine ("\nEl artista/título que buscaba está en la base de datos.\n");
            }
            else Console.WriteLine ("\nEl artista/título que buscaba NO está en la base de datos
.\n");
        }
        if ((num<1)||(num>4)){
            Console.WriteLine ("\nElija una opción del menú entre 1 y 4.\n");
        }
    } while (num != 4);

    if (num == 4) {
        Console.WriteLine ("\nEl programa ha finalizado.");
    }
}
catch (Exception error){
    Console.WriteLine ("\nHa habido un error: {0}\n", error.Message);
    goto inicio;
}
}
}
}

```

// Hecho con métodos

```
using System;
```

```
namespace Tone
```

```
{  
    public struct duracion{  
        public int minutos;  
        public int segundos;  
    }  
}
```

```
public struct cancionMP3 {  
    public string Artista;  
    public string Titulo;  
    public duracion duracion;  
    public int Tamanyo;  
}
```

```
public class doceEnero  
{  
    public static void Main()  
    {
```

```
        cancionMP3[] canciones = new cancionMP3[5];  
        int contadorCanciones = 0;  
        int num = 0;
```

```
        do{
```

```
            Console.Write ("\n\tMENÚ DE PROGRAMA\n\n1) Añadir cancion.\n2) Mostrar  
titulos.\n3) Buscar artista/titulo.\n4) Salir.");  
            Console.Write ("\n¿Qué desea hacer?: ");  
            num = Convert.ToInt32 (Console.ReadLine ());
```

```
            switch(num){
```

```
            case 1:
```

```
                cancionMP3 nuevaCancion = pedirCancion();  
                canciones[contadorCanciones] = nuevaCancion;  
                contadorCanciones++;  
                break;
```

```
            case 2:
```

```
                mostrarTitulo(canciones);  
                break;
```

```
            case 3:
```

```
                string cancionEncontrada = buscarPorTituloArtista(canciones);  
                if(cancionEncontrada!=null)
```

```
                    Console.WriteLine("La canción que busca tiene el título {0}", cancionEncontrada);
```

```

ada);
        else Console.WriteLine("No la hemos encontrado, pregúntale a tu mujer");
        break;
    }
}
while (num!=4);

if (num == 4) {
    Console.Write ("El programa ha finalizado.");
}

Console.Read ();

}

public static cancionMP3 pedirCancion(){
    cancionMP3 unaCancion;

    Console.Write ("Introduce el artista: ");
    unaCancion.Artista = Console.ReadLine ();

    Console.Write ("Introduce el título: ");
    unaCancion.Titulo = Console.ReadLine ();

    Console.Write ("Introduce la duración (minutos): ");
    unaCancion.duracion.minutos =Convert.ToInt32(Console.ReadLine ());

    Console.Write ("Introduce la duración (segundos): ");
    unaCancion.duracion.segundos =Convert.ToInt32(Console.ReadLine ());

    Console.Write ("Introduce el tamaño en KB: ");
    unaCancion.Tamanyo =Convert.ToInt32(Console.ReadLine ());

    return unaCancion;
}

public static void mostrarTitulo(cancionMP3[] canciones){
    for (int i = 0; i < canciones.Length; i++) {
        Console.Write (canciones[i].Titulo + " ");
    }
    Console.WriteLine ();
}

public static string buscarPorTituloArtista(cancionMP3[] todasLasCanciones){
    string textoParaBuscar = Console.ReadLine();

```

```

string tituloCancion = null;

for (int i = 0; i < todasLasCanciones.Length; i++) {
    if ( (todasLasCanciones [i].Titulo != null) &&
        (todasLasCanciones [i].Titulo.IndexOf (textoParaBuscar) != -1
         || todasLasCanciones [i].Artista.IndexOf (textoParaBuscar) != -1)) {

        tituloCancion = todasLasCanciones [i].Titulo;
    }
}
return tituloCancion;
}
}
}

```

(4.4.3.1) Un programa que te pida tu nombre y lo muestre en pantalla separando cada letra de la siguiente con un espacio. Por ejemplo, si tu nombre es "Juan", debería aparecer en pantalla "J u a n".

```

using System;

public class cadenas
{
    public static void Main()
    {
        string nombre;

        Console.Write ("Introduce tu nombre: ");
        nombre = Console.ReadLine ();

        for (int i = 0; i < nombre.Length; i++) {
            Console.Write (nombre [i] + " ");
        }
    }
}

```

(4.4.3.2) Un programa que pida una frase al usuario y la muestre en orden inverso (de la última letra a la primera).

```
using System;

public class cadenas
{
    public static void Main()
    {
        string frase;

        Console.Write ("Introduce una frase: ");
        frase = Console.ReadLine ();

        for (int i=frase.Length-1; i >=0; i--) {
            Console.Write (frase [i]);
        }
    }
}
```

(4.4.4.2) Un programa que te pida tu nombre y lo muestre en pantalla como un triángulo creciente. Por ejemplo, si tu nombre es "Juan", debería aparecer en pantalla:

```
J
Ju
Jua
Juan
```

```
using System;

public class cadenas
{
    public static void Main()
    {
        string nombre, subcadena;

        Console.Write ("Introduce tu nombre: ");
        nombre = Console.ReadLine ();

        for (int i = 1; i <= nombre.Length; i++) {
            subcadena = nombre.Substring (0, i);

            Console.WriteLine (subcadena);
            Console.WriteLine ();
        }
    }
}
```

(4.4.5.1) Un programa que pida al usuario 10 frases, las guarde en un array, y luego le pregunte textos de forma repetitiva, e indique si cada uno de esos textos aparece como parte de alguno de los elementos del array. Terminará cuando el texto introducido sea "fin".

```
using System;
```

```
public class index
```

```
{
    public static void Main ()
    {
        int contador=1;
        string[] frase= new string[5];
        string palabra="";
        bool encontrada = false;

        do {

            for (int i = 0; i < frase.Length; i++) {
                Console.Write ("Introduce la {0}ª frase: ", contador);
                frase [i] = Console.ReadLine ();
                contador++;
            }
        } while (contador < 6);

        do {
            string palabraContenedora = "";
            encontrada=false;
            Console.Write ("\nIntroduce la palabra que quieres buscar: ");
            palabra = Console.ReadLine ();
            if (palabra=="fin") break;
            for (int i = 0; i < frase.Length; i++) {
                if (frase[i].IndexOf (palabra) != -1) {
                    encontrada = true;
                    palabraContenedora=frase[i];
                    break;
                }
            }
            if (encontrada == true) {
                Console.Write ("\nLa palabra se encuentra en la frase {0}.", palabraContenedora)
;
            }
            else {
                Console.Write ("\nLa palabra introducida NO se encuentra en ninguna frase.");
            }
        } while (palabra != "fin");
    }
}
```

```

        Console.WriteLine ("\nFin de programa.");
    }
}

```

/ Con LastIndexOf es igual y Contains devuelve un "true" o "false", es decir, if (frase[i].Contains (palabra) == true) */*

(4.4.5.2) Crea una versión del ejercicio 4.4.5.1 en el que, en caso de que alguno de los textos aparezca como subcadena, se avise además si se encuentra exactamente al principio.

```
using System;
```

```

public class index
{
    public static void Main ()
    {
        int contador=1;
        string[] frase= new string[5];
        string palabra="";
        bool encontrada = false;

        do {

            for (int i = 0; i < frase.Length; i++) {
                Console.WriteLine ("Introduce la {0}ª frase: ", contador);
                frase [i] = Console.ReadLine ();
                contador++;
            }
        } while (contador < 6);

        do {
            string palabraContenedora = "";
            encontrada = false;
            bool estaAlPrincipio = false;

            Console.WriteLine ("\nIntroduce la palabra que quieres buscar: ");
            palabra = Console.ReadLine ();
            if (palabra == "fin")
                break;
            for (int i = 0; i < frase.Length; i++) {
                int indiceDeLaPalabra = frase [i].IndexOf (palabra);
                if (indiceDeLaPalabra != -1) {
                    encontrada = true;
                    palabraContenedora = frase [i];
                }
            }
        } while (!encontrada);

        Console.WriteLine ("\nLa palabra encontrada es: {0}", palabraContenedora);
    }
}

```



```

        if (indiceDeLaPalabra == 0) {
            estaAlPrincipio = true;
        }
        break;
    }
}
if (encontrada) {
    if (estaAlPrincipio) {
        Console.WriteLine("\nLa cadena está al principio de la palabra: " + palabraConten
edora);
    } else {
        Console.WriteLine("\nLa palabra se encuentra en alguna posición de la palabra: "
+ palabraContenedora);
    }
} else {
    Console.WriteLine("\nLa palabra introducida NO se encuentra en ninguna frase.");
}
} while (palabra != "fin");

Console.WriteLine("\nFin de programa.");
}
}

```

(4.4.6.1) Una variante del ejercicio 4.4.5.2, que no distinga entre mayúsculas y minúsculas a la hora de buscar.

```
using System;
```

```
public class index
```

```
{
```

```
    public static void Main ()
```

```
    {
```

```
        int contador=1;
```

```
        string[] frase= new string[5];
```

```
        string palabra="";
```

```
        bool encontrada = false;
```

```
        do {
```

```
            for (int i = 0; i < frase.Length; i++) {
```

```
                Console.Write ("Introduce la {0}ª frase: ", contador);
```

```
                frase [i] = Console.ReadLine ();
```

```
                contador++;
```

```
            }
```

```
        } while (contador < 6);
```

```
        do {
```

```
            string palabraContenedora = "";
```

```
            encontrada = false;
```

```
            bool estaAlPrincipio = false;
```

```
            Console.Write ("\nIntroduce la palabra que quieres buscar: ");
```

```
            palabra = Console.ReadLine ();
```

```
            if (palabra.ToLower() == "fin")
```

```
                break;
```

```
            for (int i = 0; i < frase.Length; i++) {
```

```
                int indiceDeLaPalabra = frase [i].ToUpper().IndexOf (palabra.ToUpper());
```

```
                if (indiceDeLaPalabra != -1) {
```

```
                    encontrada = true;
```

```
                    palabraContenedora = frase [i];
```

```
                    if (indiceDeLaPalabra == 0) {
```

```
                        estaAlPrincipio = true;
```

```
                    }
```

```
                break;
```

```
            }
```

```
        }
```

```
        if (encontrada) {
```

```
            if (estaAlPrincipio) {
```

```
                Console.Write ("\nLa cadena esta al principio de la palabra: " + palabraConten
```

```
edora);
```

```

        } else {
            Console.WriteLine("\nLa palabra se encuentra en alguna posicion de la palabra: "
+ palabraContenedora);
        }
    } else {
        Console.WriteLine("\nLa palabra introducida NO se encuentra en ninguna frase.");
    }
} while (true); //también se suele poner while (1==1) para hacer que el bucle sea infinito hasta que llega al break de la línea 29

Console.WriteLine("\nFin de programa.");
}
}

```

Otra manera de hacerlo:

```

using System;

public class mayúsculas
{
    public static void Main ()
    {
        string[] palabra=new string[5];
        string buscador;
        bool encontradaPrincipio, encontrada;
        int posicion=0, posicion2=0;

        for (int i = 0; i < palabra.Length; i++) {
            Console.WriteLine ("Introduce la {0}ª palabra: ", i+1);
            palabra[i] = Console.ReadLine ();
        }

        do {
            Console.WriteLine ("\nEscribe la palabra que quieres buscar: ");
            buscador = Console.ReadLine ();

            if (buscador.ToLower() == "fin")
                break;

            encontradaPrincipio = false;
            encontrada=false;

            for (int i = 0; i < palabra.Length; i++) {
                if ((palabra [i].IndexOf (buscador.ToLower())!=0) && (palabra [i].IndexOf (buscador.ToLower())==0)){
                    encontradaPrincipio = true;
                    posicion=i;
                }
            }
        } while (true);

        Console.WriteLine ("\nLa palabra buscada se encuentra en la posición: " + posicion);
    }
}

```

```

        break;
    }
    if ((palabra [i].IndexOf (buscador.ToLower())!=0)
1) && (palabra [i].IndexOf (buscador.ToLower())!=0)) {
        encontrada = true;
        posicion=i;
        posicion2=palabra[i].IndexOf(buscador.ToLower());
        break;
    }
}
if (encontradaPrincipio) {
    Console.Write ("El fragmento buscado se encuentra al inicio de la posición {0}
.", posicion+1);
} else {

    if (encontrada){
        Console.Write ("El fragmento se encuentra en la posición {0}, pero no al princ
ipio sino en la posición {1} de la palabra.", posicion+1, posicion2+1);
    } else {
        Console.Write ("La palabra NO se encuentra en el array.");
    }
}
} while (1 == 1);

Console.WriteLine ("El programa ha finalizado.");
}
}

```

(4.4.6.2) Un programa que pida al usuario una frase y elimine todos los espacios redundantes que contenga (debe quedar sólo un espacio entre cada palabra y la siguiente).

```

using System;

public class replace
{
    public static void Main ()
    {
        string frase;

        Console.Write ("Introduce una frase: ");
        frase = Console.ReadLine();
        Console.WriteLine (frase.Replace(" ", " "));
    }
}

```

(4.4.7.1) Un programa que pida al usuario una frase y muestre sus palabras en orden inverso.

```
using System;
```

```
public class replace
```

```
{
```

```
    public static void Main ()
```

```
    {
```

```
        string frase;
```

```
        char[] delimitador = {' '};
```

```
        Console.Write ("Introduzca una frase: ");
```

```
        frase = Console.ReadLine ();
```

```
        string[] fragmentos = frase.Split (delimitador);
```

```
        int contador = 1;
```

```
        for (int i = fragmentos.Length-1; i >=0; i--) {
```

```
            Console.WriteLine ("{0}º palabra: {1}", contador, fragmentos [i]);
```

```
            contador++;
```

```
        }
```

```
    }
```

```
}
```

(4.4.7.2) Un programa que pida al usuario varios números separados por espacios y muestre su suma.

```
using System;

public class split
{
    public static void Main ()
    {
        string numeros;
        char[] delimitador = { ' ' };
        int suma = 0;

        Console.Write ("Introduce varios números separados por espacios: ");
        numeros = Console.ReadLine ();

        string[] fragmentos = numeros.Split (delimitador);

        for (int i = 0; i < fragmentos.Length; i++) {
            if (fragmentos [i] != "") {
                suma += Convert.ToInt32 (fragmentos[i]);
            }
        }
        Console.Write ("La suma de los números introducidos es: {0}", suma);
    }
}
```

(4.4.8.2) Un programa que pida al usuario cinco frases, las guarde en un array y muestre la "mayor" de ellas (la que aparecería en último lugar en un diccionario).

```
using System;
```

```
public class espacios
```

```
{  
    public static void Main ()  
    {  
        string[] palabra = new string[5];  
        string comparadora;  
  
        for (int i = 0; i < palabra.Length; i++) {  
            Console.Write ("Introduce una palabra: ");  
            palabra[i] = Console.ReadLine ();  
        }  
  
        comparadora = palabra [0];  
        for (int i = 1; i < palabra.Length; i++) {//las siguientes dos opciones son válidas  
            /*if (palabra [i].CompareTo (comparadora) > 0) {*/  
            if (string.Compare(palabra [i],comparadora,true) > 0) {  
                comparadora = palabra [i];  
            }  
        }  
        Console.Write ("La mayor palabra en orden alfabético es: {0}", comparadora);  
    }  
}
```

(4.4.9.2) Un programa que pida una cadena al usuario y la modifique, de modo que las letras de las posiciones impares (primera, tercera, etc.) estén en minúsculas y las de las posiciones pares estén en mayúsculas, mostrando el resultado en pantalla. Por ejemplo, a partir de un nombre como "Nacho", la cadena resultante sería "nAcHo".

```
using System;
using System.Text;

public class Builder
{
    public static void Main ()
    {
        string nombre;
        StringBuilder cadena = new StringBuilder ();

        Console.Write ("Introduce tu nombre: ");
        nombre = Console.ReadLine ();

        string mayusculas = nombre.ToUpper ();
        string minusculas = nombre.ToLower ();

        for (int i = 0; i < nombre.Length; i++) {
            if (i % 2 == 0) {
                cadena.Append (minusculas [i]);
            } else {
                cadena.Append (mayusculas [i]);
            }
        }
        Console.Write (cadena.ToString());
    }
}
```


(4.4.9.3) Crear un juego del ahorcado, en el que un primer usuario introduzca la palabra a adivinar, se muestre esta programa oculta con guiones (-----) y el programa acepte las letras que introduzca el segundo usuario, cambiando los guiones por letras correctas cada vez que acierte (por ejemplo, a---a-t-). La partida terminará cuando se acierte la palabra por completo o el usuario agote sus 8 intentos.

```
using System;
using System.Text;

public class ahorcado
{
    public static void Main ()
    {
        Random aleatorio = new Random ();
        string[] palabras = { "el gran lewovsky", "carlos", "rosa del desierto", "canela en rama",
        "clara de huevo", "pizpireta", "rodrigo diaz de vivar", "gonzalo", "javier", "noelia" };
        string elegida;
        int numeroAzar;

        // El ordenador elige una palabra al azar

        numeroAzar = aleatorio.Next (0,palabras.Length);
        elegida=palabras[numeroAzar];

        // Se sustituyen las letras por guiones

        StringBuilder nuevaCadena = new StringBuilder ();
        foreach (char letra in elegida) {
            if (letra == ' ') {
                nuevaCadena.Append (' ');
            } else {
                nuevaCadena.Append ('-');
            }
        }

        //Más variables

        int fallosRestantes = 8;
        char letraUsuario;
        bool terminado = false;

        //Bucle de repetición

        do {
            //Se muestra en consola la palabra oculta y los fallos restantes
```

```

    Console.Write ("Palabra oculta: {0}", nuevaCadena);
    Console.WriteLine ("\n\nFallos restantes: {0}", fallosRestantes);

    //Se pide una nueva letra al usuario

    Console.Write ("Elige una letra: ");
    letraUsuario = Convert.ToChar (Console.ReadLine ());

    //Si el usuario falla, se resta una oportunidad

    if (elegida.IndexOf (letraUsuario) == -1) {
        fallosRestantes--;
    }

    //Si el usuario acierta, la letra sustituye a los guiones

    for (int i=0; i<elegida.Length; i++){
        if (letraUsuario == elegida[i]){
            nuevaCadena[i]=letraUsuario;
        }
    }

    //Si el usuario acierta todas las letras o se queda sin oportunidades se acaba la part
ida

    if (nuevaCadena.ToString().IndexOf ('-')== -1){
        Console.WriteLine ("\n\nENHORABUENA!!!! {0}", elegida.ToUpper());
        terminado=true;
    }

    if (fallosRestantes==0){
        Console.WriteLine ("Eres muuuuuuuu malo!!! La palabra era: {0}",elegida);
        terminado=true;
    }

    Console.WriteLine();

} while (!terminado);
}
}

```

(4.5.2) Un programa que pida al usuario una frase y la descomponga en subcadenas separadas por espacios, usando "Split". Luego debe mostrar cada subcadena en una línea nueva, usando "foreach".

```
using System;

public class foreacher
{
    public static void Main ()
    {
        string frase;
        string[] nuevaFrase;
        char delimitador = ' ';

        Console.Write ("Introduce una frase: ");
        frase = Console.ReadLine ();

        nuevaFrase = frase.Split (delimitador);

        foreach (string fragmento in nuevaFrase) {
            Console.WriteLine (fragmento);
        }
    }
}
```

(4.5.3) Un programa que pida al usuario varios números separados por espacios y muestre su suma (como el del ejercicio 4.4.7.2), pero empleando "foreach".

```
using System;

public class foreacher
{
    public static void Main ()
    {
        string numeros;
        int suma = 0;
        string[] nuevaFrase;
        char delimitador = ' ';

        Console.Write ("Introduce varios números: ");
        numeros = Console.ReadLine ();

        nuevaFrase = numeros.Split (delimitador);

        foreach (string numero in nuevaFrase) {
            suma += Convert.ToInt32 (numero);
        }
        Console.WriteLine ("La suma es: {0}", suma);
    }
}
```

(4.6.1) Un programa que pida el nombre, el apellido y la edad de una persona, los almacene en un "struct" y luego muestre los tres datos en una misma línea, separados por comas.

```
using System;

public class estructura
{
    struct persona
    {
        public string nombre;
        public string apellido;
        public int edad;
    }

    public static void Main ()
    {
        persona tipoPersona;

        Console.Write ("Introduce el nombre: ");
        tipoPersona.nombre = Console.ReadLine ();

        Console.Write ("Introduce el apellido: ");
        tipoPersona.apellido = Console.ReadLine ();

        Console.Write ("Introduce la edad: ");
        tipoPersona.edad = Convert.ToInt32 (Console.ReadLine ());

        Console.Write (tipoPersona.nombre + ',' + tipoPersona.apellido + ',' + tipoPersona.edad)
;
    }
}
```

(4.6.2) Un programa que pida datos de 8 personas: nombre, día de nacimiento, mes de nacimiento, y año de nacimiento (que se deben almacenar en una tabla de structs). Después deberá repetir lo siguiente: preguntar un número de mes y mostrar en pantalla los datos de las personas que cumplan los años durante ese mes. Terminará de repetirse cuando se teclee 0 como número de mes.

```
using System;
```

```
public class estructura
```

```
{
```

```
    struct nacimiento
```

```
    {
```

```
        public int dia;
```

```
        public int mes;
```

```
        public int año;
```

```
    }
```

```
    struct persona
```

```
    {
```

```
        public string nombre;
```

```
        public nacimiento fechaNacimiento;
```

```
    }
```

```
public static void Main ()
```

```
{
```

```
    persona[] tipoPersona=new persona[4];
```

```
    int mes;
```

```
    for (int i = 0; i < tipoPersona.Length; i++) {
```

```
        Console.Write ("Introduce el nombre: ");
```

```
        tipoPersona[i].nombre = Console.ReadLine ();
```

```
        Console.Write ("Introduce el día de la fecha de nacimiento: ");
```

```
        tipoPersona[i].fechaNacimiento.dia = Convert.ToInt32 (Console.ReadLine ());
```

```
        Console.Write ("Introduce el mes de la fecha de nacimiento: ");
```

```
        tipoPersona[i].fechaNacimiento.mes = Convert.ToInt32 (Console.ReadLine ());
```

```
        Console.Write ("Introduce el año de la fecha de nacimiento: ");
```

```
        tipoPersona[i].fechaNacimiento.año = Convert.ToInt32 (Console.ReadLine ());
```

```
    }
```

```
    do {
```

```
        Console.Write ("\nIntroduce un mes: ");
```

```

        mes = Convert.ToInt32 (Console.ReadLine ());
        bool encontrada = false;
        if (mes==0) break;
        for (int i=0; i<tipoPersona.Length; i++){
            if (mes==tipoPersona[i].fechaNacimiento.mes){
                encontrada=true;
                Console.WriteLine ("Datos: {0}, {1}-{2}-{3}", tipoPersona[i].nombre, tipoPersona[i].fechaNacimiento.dia, tipoPersona[i].fechaNacimiento.mes, tipoPersona[i].fechaNacimiento.año);
            }
        }
        if (!encontrada){
            Console.Write ("No hay resultados.");
        }
    } while (mes != 0);
}
}

```

(4.6.4) Mejorar la base de datos de ficheros (ejemplo 04_06a) para que no permita introducir tamaños incorrectos (números negativos) ni nombres de fichero vacíos.

```
using System;
```

```
public class ejemploCompleto
```

```

{
    struct fichero
    {
        public string nombre;
        public long tamaño;
    }

    public static void Main ()
    {
        fichero[] baseDatos=new fichero[5];
        int num;
        int fichas = 0;
        string buscar;

        do {
            Console.Write ("\n\t\tMENÚ DE PROGRAMA");
            Console.Write ("\n\n1) Añadir datos de un nuevo fichero.");
            Console.Write ("\n2) Mostrar los nombres de todos los ficheros almacenados.");
            Console.Write ("\n3) Mostrar ficheros que sean de más de un cierto tamaño (por ejemplo, 2000 KB).");
            Console.Write ("\n4) Ver todos los datos de un cierto fichero (a partir de su nombre)

```

```

.");
    Console.WriteLine ("\n5) Salir de la aplicación.");

    Console.WriteLine ("\n\nElija una opción del menú: ");
    num = Convert.ToInt32 (Console.ReadLine ());

    switch (num) {
    case 1:
        if (fichas < 5) {
            do {
                Console.WriteLine ("Introduzca el nombre del fichero: ");
                baseDatos [fichas].nombre = Console.ReadLine ();
            } while (baseDatos [fichas].nombre=="");
            do {
                Console.WriteLine ("\nIntroduzca el tamaño del fichero: ");
                baseDatos [fichas].tamaño = Convert.ToInt64 (Console.ReadLine ());
                if (baseDatos [fichas].tamaño<0)
                    Console.WriteLine ("El tamaño tiene que ser positivo.");
            } while (baseDatos [fichas].tamaño<0);
            fichas++;
        } else {
            Console.WriteLine ("\nNo se pueden almacenar más. La base de datos está llena.\n"
);
        }
        break;
    case 2:
        for (int i = 0; i < baseDatos.Length; i++) {
            if (baseDatos [i].nombre != null) {
                Console.WriteLine ("-
En la {0}º posición de la base de datos se encuentra el fichero: {1}", i+1, baseDatos [i].no
mbre);
            }
        }
        break;
    case 3:
        for (int i=0; i<baseDatos.Length; i++){
            if (baseDatos[i].tamaño>2000){
                Console.WriteLine ("-
En la {0}º posición se encuentra el fichero \"{1}\" con el tamaño {2}.", i+1, baseDatos[i].n
ombre, baseDatos[i].tamaño);
            }
        }
        break;
    case 4:
        Console.WriteLine ("Introduzca el fichero que quiere buscar: ");
        buscar=Console.ReadLine();

```



```

        for (int i = 0; i < baseDatos.Length; i++) {
            if (baseDatos[i].nombre == buscar) {
                Console.WriteLine ("-
En la {0}º posición se encuentra el fichero \"{1}\" con el tamaño: {2}.", i+1, baseDatos[i].
nombre, baseDatos[i].tamaño);
            }
        }
        break;
    }
} while (num != 5);
Console.Write ("\n\nEl programa ha finalizado.");
}
}

```

(4.6.5) Ampliar la base de datos de ficheros (ejemplo 46) para que incluya una opción de búsqueda parcial, en la que el usuario indique parte del nombre y se muestre todos los ficheros que contienen ese fragmento (usando "Contains" o "IndexOf"). Esta búsqueda no debe distinguir mayúsculas y minúsculas (con la ayuda de ToUpper o ToLower).

```

using System;

public class ejemplouy
{
    struct fichero
    {
        public string nombre;
        public long tamaño;
    }

    public static void Main ()
    {
        fichero[] baseDatos=new fichero[5];
        int num;
        int fichas = 0;
        string buscar;

        do {
            Console.Write ("\n\t\tMENÚ DE PROGRAMA");
            Console.Write ("\n\n1) Añadir datos de un nuevo fichero.");
            Console.Write ("\n2) Mostrar los nombres de todos los ficheros almacenados.");
            Console.Write ("\n3) Mostrar ficheros que sean de más de un cierto tamaño (por eje
mplo, 2000 KB).");

```

```

        Console.WriteLine("\n4) Ver todos los datos de un cierto fichero (a partir de su nombre)");
    };

    Console.WriteLine("\n5) Salir de la aplicación.");

    Console.WriteLine("\n\nElija una opción del menú: ");
    num = Convert.ToInt32(Console.ReadLine());

    switch (num) {
    case 1:
        if (fichas < 5) {
            do {
                Console.WriteLine("Introduzca el nombre del fichero: ");
                baseDatos[fichas].nombre = Console.ReadLine();
            } while (baseDatos[fichas].nombre == "");
            do {
                Console.WriteLine("\nIntroduzca el tamaño del fichero: ");
                baseDatos[fichas].tamaño = Convert.ToInt64(Console.ReadLine());
                if (baseDatos[fichas].tamaño < 0)
                    Console.WriteLine("El tamaño tiene que ser positivo.");
            } while (baseDatos[fichas].tamaño < 0);
            fichas++;
        } else {
            Console.WriteLine("\nNo se pueden almacenar más. La base de datos está llena.\n");
        }
        break;
    case 2:
        for (int i = 0; i < baseDatos.Length; i++) {
            if (baseDatos[i].nombre != null) {
                Console.WriteLine("-
En la {0}ª posición de la base de datos se encuentra el fichero: {1}", i+1, baseDatos[i].nombre);
            }
        }
        break;
    case 3:
        for (int i = 0; i < baseDatos.Length; i++) {
            if (baseDatos[i].tamaño > 2000) {
                Console.WriteLine("-
En la {0}ª posición se encuentra el fichero \"{1}\" con el tamaño {2}.", i+1, baseDatos[i].nombre, baseDatos[i].tamaño);
            }
        }
        break;
    case 4:
        bool encontrada = false;

```

```

        string palabraEncontrada="";
        Console.Write ("Introduzca el fichero que quiere buscar: ");
        buscar=Console.ReadLine();
        buscar=buscar.ToUpper();
        for (int i = 0; i < baseDatos.Length; i++) {
            if (baseDatos [i].nombre!=null && baseDatos [i].nombre.ToUpper().IndexOf (
buscar)!=-1) {
                encontrada=true;
                palabraEncontrada=baseDatos [i].nombre;
            }
        }
        if (encontrada){
            Console.WriteLine ("Se ha encontrado en el fichero {0}", palabraEncontrada);
        } else {
            Console.Write ("No hay resultados.");
        }
        break;
    }
} while (num != 5);
Console.Write ("\n\nEl programa ha finalizado.");
}
}

```

(4.6.7) Ampliar la base de datos de ficheros (ejemplo 04_06a) para que se pueda borrar un cierto dato (habrá que "mover hacia atrás" todos los datos que había después de ese, y disminuir el contador de la cantidad de datos que tenemos).

```
using System;
```

```

public class ejemploCompleto
{
    struct fichero
    {
        public string nombre;
        public long tamaño;
    }

    public static void Main ()
    {
        fichero[] baseDatos=new fichero[5];
        int num;
        int fichas = 0;
        string buscar;

        do {

```

```

Console.Write ("\t\tMENÚ DE PROGRAMA");
Console.Write ("\n\n1) Añadir datos de un nuevo fichero.");
Console.Write ("\n2) Mostrar los nombres de todos los ficheros almacenados.");
Console.Write ("\n3) Mostrar ficheros que sean de más de un cierto tamaño (por ejemplo, 2000 KB).");
Console.Write ("\n4) Ver todos los datos de un cierto fichero (a partir de su nombre).");
Console.Write ("\n5) Borrar un fichero.");
Console.Write ("\n6) Salir de la aplicación.");

```

```

Console.Write ("\n\nElija una opción del menú: ");
num = Convert.ToInt32 (Console.ReadLine ());

```

```

switch (num) {
case 1:
    if (fichas < 5) {
        do {
            Console.Write ("Introduzca el nombre del fichero: ");
            baseDatos [fichas].nombre = Console.ReadLine ();
        } while (baseDatos [fichas].nombre=="");
        do {
            Console.Write ("\nIntroduzca el tamaño del fichero: ");
            baseDatos [fichas].tamaño = Convert.ToInt64 (Console.ReadLine ());
            if (baseDatos [fichas].tamaño<0)
                Console.Write ("El tamaño tiene que ser positivo.");
        } while (baseDatos [fichas].tamaño<0);
        fichas++;
    } else {
        Console.Write ("\nNo se pueden almacenar más. La base de datos está llena.\n");
    }
};

```

```

    }
    break;
case 2:
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre != null) {
            Console.WriteLine ("-
En la {0}º posición de la base de datos se encuentra el fichero: {1}", i+1, baseDatos [i].nombre);
        }
    }
    break;

```

```

case 3:
    for (int i=0; i<baseDatos.Length; i++){
        if (baseDatos[i].tamaño>2000){
            Console.WriteLine ("-
En la {0}º posición se encuentra el fichero \"{1}\" con el tamaño {2}.", i+1, baseDatos[i].nombre, baseDatos[i].tamaño);
        }
    }
    break;

```

```

ombre, baseDatos[i].tamaño);
    }
}
break;
case 4:
    bool encontrada=false;
    string palabraEncontrada="";
    Console.Write ("Introduzca el fichero que quiere buscar: ");
    buscar=Console.ReadLine();
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre.IndexOf (buscar)!=-1) {
            encontrada=true;
            palabraEncontrada=baseDatos [i].nombre;
        }
    }
    if (encontrada){
        Console.WriteLine ("Se ha encontrado en el fichero {0}", palabraEncontrada);
    } else {
        Console.Write ("No hay resultados.");
    }
    break;
case 5:
    int posicion;
    Console.Write("Introduzca la posición del dato que desea borrar: ");
    posicion=Convert.ToInt32 (Console.ReadLine());
    for (int i=posicion; i<baseDatos.Length-1; i++){
        baseDatos[i]=baseDatos[i+1];
    }
    fichas--;
    break;
}
} while (num != 6);
Console.Write ("\n\nEl programa ha finalizado.");
}
}

```

(4.6.8) Mejorar la base de datos de ficheros (ejemplo 04_06a) para que se pueda modificar un cierto dato a partir de su número (por ejemplo, el dato número 3). En esa modificación, se deberá permitir al usuario pulsar Intro sin teclear nada, para indicar que no desea modificar un cierto dato, en vez de reemplazarlo por una cadena vacía.

```
using System;
```

```
public class ejemploCompleto
```

```
{  
    struct fichero  
    {  
        public string nombre;  
        public long tamaño;  
    }  
  
    public static void Main ()  
    {  
        fichero[] baseDatos = new fichero[5];  
        int num;  
        int fichas = 0;  
        string buscar;  
  
        do {  
            Console.Write ("\n\t\tMENÚ DE PROGRAMA");  
            Console.Write ("\n\n1) Añadir datos de un nuevo fichero.");  
            Console.Write ("\n2) Mostrar los nombres de todos los ficheros almacenados.");  
            Console.Write ("\n3) Mostrar ficheros que sean de más de un cierto tamaño (por ejemplo, 2000 KB).");  
            Console.Write ("\n4) Ver todos los datos de un cierto fichero (a partir de su nombre).");  
            Console.Write ("\n5) Borrar un fichero.");  
            Console.Write ("\n6) Modificar un fichero.");  
            Console.Write ("\n7) Salir de la aplicación.");  
  
            Console.Write ("\n\nElija una opción del menú: ");  
            num = Convert.ToInt32 (Console.ReadLine ());  
  
            switch (num) {  
                case 1:  
                    if (fichas < 5) {  
                        do {  
                            Console.Write ("Introduzca el nombre del fichero: ");  
                            baseDatos [fichas].nombre = Console.ReadLine ();  
                        } while (baseDatos [fichas].nombre == "");  
                        do {  
                            Console.Write ("\nIntroduzca el tamaño del fichero: ");
```

```

        baseDatos [fichas].tamaño = Convert.ToInt64 (Console.ReadLine ());
        if (baseDatos [fichas].tamaño < 0)
            Console.Write ("El tamaño tiene que ser positivo.");
        } while (baseDatos [fichas].tamaño < 0);
        fichas++;
    } else {
        Console.Write ("\nNo se pueden almacenar más. La base de datos está llena.\n"
    );
    }
    break;
case 2:
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre != null) {
            Console.WriteLine ("-
En la {0}º posición de la base de datos se encuentra el fichero: {1}", i + 1, baseDatos [i].no
mbre);
        }
    }
    break;
case 3:
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].tamaño > 2000) {
            Console.WriteLine ("-
En la {0}º posición se encuentra el fichero \"{1}\" con el tamaño {2}.", i + 1, baseDatos [i]
.nombre, baseDatos [i].tamaño);
        }
    }
    break;
case 4:
    bool encontrada = false;
    string palabraEncontrada = "";
    Console.Write ("Introduzca el fichero que quiere buscar: ");
    buscar = Console.ReadLine ();
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre.IndexOf (buscar) != -1) {
            encontrada = true;
            palabraEncontrada = baseDatos [i].nombre;
        }
    }
    if (encontrada) {
        Console.WriteLine ("Se ha encontrado en el fichero {0}", palabraEncontrada);
    } else {
        Console.Write ("No hay resultados.");
    }
    break;
case 5:
    int posicion;

```

```

        Console.WriteLine("Introduzca la posición del dato que desea borrar: ");
        posicion = Convert.ToInt32(Console.ReadLine());
        for (int i = posicion; i < baseDatos.Length - 1; i++) {
            baseDatos[i] = baseDatos[i + 1];
        }
        fichas--;
        break;
    case 6:
        int indice;
        Console.WriteLine("Introduzca la posición del archivo a modificar: ");
        indice = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Nuevo nombre del fichero " + baseDatos[indice].nombre + ": ");
        string nuevoNombre = Console.ReadLine();
        if(nuevoNombre != ""){
            baseDatos[indice].nombre = nuevoNombre;
        }

        Console.WriteLine("Nuevo tamaño del fichero:");
        string nuevoTamanio = Console.ReadLine();
        if(nuevoTamanio!=""){
            baseDatos[indice].tamaño = Convert.ToInt64(nuevoTamanio);
        }
        break;
    default:
        Console.WriteLine("Introduzca un número entre 1 y 7.");
        break;
    }
} while (num != 7);
Console.WriteLine("\n\nEl programa ha finalizado.");
}
}

```


(4.6.9) Ampliar la base de datos de ficheros (ejemplo 04_06a) para que se permita ordenar los datos por nombre. Para ello, deberás buscar información sobre algún método de ordenación sencillo, como el "método de burbuja" (en el siguiente apartado tienes algunos), y aplicarlo a este caso concreto.

```
using System;
```

```
public class ejemploCompleto
```

```
{  
    struct fichero  
    {  
        public string nombre;  
        public long tamaño;  
    }  
  
    public static void Main ()  
    {  
        fichero[] baseDatos = new fichero[5];  
        int num;  
        int fichas = 0;  
        string buscar;  
  
        do {  
            Console.WriteLine("\t\tMENÚ DE PROGRAMA");  
            Console.WriteLine("\n1) Añadir datos de un nuevo fichero.");  
            Console.WriteLine("\n2) Mostrar los nombres de todos los ficheros almacenados.");  
            Console.WriteLine("\n3) Mostrar ficheros que sean de más de un cierto tamaño (por ejemplo, 2000 KB).");  
            Console.WriteLine("\n4) Ver todos los datos de un cierto fichero (a partir de su nombre).");  
            Console.WriteLine("\n5) Borrar un fichero.");  
            Console.WriteLine("\n6) Modificar un fichero.");  
            Console.WriteLine("\n7) Ordenar por nombre.");  
            Console.WriteLine("\n8) Salir de la aplicación.");  
  
            Console.WriteLine("\nElija una opción del menú: ");  
            try{  
                num = Convert.ToInt32(Console.ReadLine());  
            }  
            catch(Exception Exception){  
                num = 99999;  
            }  
  
            switch (num) {  
                case 1:  
                    if (fichas < 5) {
```

```

do {
    Console.Write ("Introduzca el nombre del fichero: ");
    baseDatos [fichas].nombre = Console.ReadLine ();
} while (baseDatos [fichas].nombre == "");
do {
    Console.Write ("\nIntroduzca el tamaño del fichero: ");
    baseDatos [fichas].tamaño = Convert.ToInt64 (Console.ReadLine ());
    if (baseDatos [fichas].tamaño < 0)
        Console.Write ("El tamaño tiene que ser positivo.");
} while (baseDatos [fichas].tamaño < 0);
fichas++;
} else {
    Console.Write ("\nNo se pueden almacenar más. La base de datos está llena.\n"
);
}
break;
case 2:
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre != null) {
            Console.WriteLine ("-
En la {0}ª posición de la base de datos se encuentra el fichero: {1}", i + 1, baseDatos [i].no
mbre);
        }
    }
    break;
case 3:
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].tamaño > 2000) {
            Console.WriteLine ("-
En la {0}ª posición se encuentra el fichero \"{1}\" con el tamaño {2}.", i + 1, baseDatos [i]
.nombre, baseDatos [i].tamaño);
        }
    }
    break;
case 4:
    bool encontrada = false;
    string palabraEncontrada = "";
    Console.Write ("Introduzca el fichero que quiere buscar: ");
    buscar = Console.ReadLine ();
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre.IndexOf (buscar) != -1) {
            encontrada = true;
            palabraEncontrada = baseDatos [i].nombre;
        }
    }
    if (encontrada) {
        Console.WriteLine ("Se ha encontrado en el fichero {0}", palabraEncontrada);
    }
}

```

```

    } else {
        Console.Write ("No hay resultados.");
    }
    break;
case 5:
    int posicion;
    Console.Write ("Introduzca la posición del dato que desea borrar: ");
    posicion = Convert.ToInt32 (Console.ReadLine ());
    for (int i = posicion; i < baseDatos.Length - 1; i++) {
        baseDatos [i] = baseDatos [i + 1];
    }
    fichas--;
    break;
case 6:
    int indice;
    Console.Write ("Introduzca la posición del archivo a modificar: ");
    indice = Convert.ToInt32 (Console.ReadLine ());

    Console.Write ("Nuevo nombre del fichero " + baseDatos[indice].nombre + ": ");
    string nuevoNombre = Console.ReadLine ();
    if(nuevoNombre != ""){
        baseDatos[indice].nombre = nuevoNombre;
    }

    Console.Write("Nuevo tamaño del fichero:");
    string nuevoTamanio = Console.ReadLine ();
    if(nuevoTamanio!=""){
        baseDatos[indice].tamaño = Convert.ToInt64 (nuevoTamanio);
    }
    break;
case 7:
    // BURBUJA
    // (Intercambiar cada pareja consecutiva que no esté ordenada)
    // Para i=1 hasta n-1
    // Para j=i+1 hasta n
    // Si A[i] > A[j]
    // Intercambiar ( A[i], A[j])
    for(int i=0; i < baseDatos.Length -1; i++) {
        for(int j=i+1; j < baseDatos.Length; j++) {

            if (baseDatos[i].nombre.CompareTo(baseDatos[j].nombre) > 0) {
                fichero datoTemporal = baseDatos[i];
                baseDatos[i] = baseDatos[j];
                baseDatos[j] = datoTemporal;
            }
        }
    }
}

```

```
        Console.Write("Base de datos ordenada.");  
        break;  
  
    default:  
        Console.WriteLine("Introduce una opción válida");  
        break;  
    }  
  
    } while (num != 8);  
    Console.Write ("\n\nEl programa ha finalizado.");  
}  
}
```