

EJERCICIOS TEMA 8 “MANEJO DE FICHEROS”

(8.1.1) Crea un programa que vaya leyendo las frases que el usuario teclea y las guarde en un fichero de texto llamado "registroDeUsuario.txt". Terminará cuando la frase introducida sea "fin" (esa frase no deberá guardarse en el fichero).

```
using System;
using System.IO;

namespace EjerciciosTema8_Ficheros_
{
    class Program
    {
        static void Main()
        {
            StreamWriter fichero = new StreamWriter("registroDeUsuario.txt");
            string frase;

            do
            {
                Console.Write("Introduce una frase en el fichero: ");
                frase = Console.ReadLine();
                if (frase != "fin")
                {
                    fichero.WriteLine(frase);
                }
            } while (frase != "fin");

            fichero.Close();
        }
    }
}
```

(8.2.1) Crea un programa que lea las tres primeras líneas del fichero creado en el ejercicio 8.1.1 y las muestre en pantalla. Usa OpenText para abrirlo.

```
using System;
using System.IO;

namespace _8._2._1
{
    class Program
    {
        static void Main()
        {
            StreamReader fichero;
            fichero = File.OpenText
("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
"/EjerciciosTema8(Ficheros)/bin/Debug/registroDeUsuario.txt");
            string linea = fichero.ReadLine();
            Console.WriteLine(linea);
            Console.WriteLine(fichero.ReadLine());
            Console.WriteLine(fichero.ReadLine());
            fichero.Close();
            Console.ReadLine();
        }
    }
}
```

(8.2.2) Crea una versión alternativa del ejercicio 8.2.1, usando el constructor de StreamReader.

```
using System;
using System.IO;

namespace _8._2._2
{
    class Program
    {
        static void Main()
        {
            StreamReader fichero = new
StreamReader("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
            "/EjerciciosTema8(Ficheros)/bin/Debug/registroDeUsuario.txt");
            string linea = fichero.ReadLine();
            Console.WriteLine(linea);
            Console.WriteLine(fichero.ReadLine());
            Console.WriteLine(fichero.ReadLine());
            fichero.Close();
            Console.ReadLine();
        }
    }
}
```

(8.2.3) Crea una versión alternativa del ejercicio 8.2.2, empleando la sintaxis alternativa de "using".

```
using System;
using System.IO;

namespace _8._2._3
{
    class Program
    {
        static void Main()
        {
            using (StreamReader fichero = new
StreamReader("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
            "/EjerciciosTema8(Ficheros)/bin/Debug/registroDeUsuario.txt"))
            {
                string linea = fichero.ReadLine();
                Console.WriteLine(linea);
                Console.WriteLine(fichero.ReadLine());
                Console.WriteLine(fichero.ReadLine());
            }
            Console.ReadLine();
        }
    }
}
```

(8.3.1) Crea una variante del ejemplo 08_03a, empleando un constructor en vez de "File.OpenText".

```
using System;
using System.IO;
```

```

namespace _8._3._1
{
    class Program
    {
        static void Main()
        {
            StreamReader fichero = new
StreamReader("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
            "/EjerciciosTema8(Ficheros)/bin/Debug/registroDeUsuario.txt");
            string linea;
            do
            {
                linea = fichero.ReadLine();
                Console.WriteLine(linea);
            } while (linea != null);
            fichero.Close();
            Console.ReadLine();
        }
    }
}

```

(8.3.2) Crea una variante del ejemplo 08_03a, empleando "using" en vez de "Close".

```

using System;
using System.IO;

namespace _8._3._2
{
    class Program
    {
        static void Main()
        {
            using (StreamReader fichero = new
StreamReader("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
            "/EjerciciosTema8(Ficheros)/bin/Debug/registroDeUsuario.txt"))
            {
                string linea;

                do
                {
                    linea = fichero.ReadLine();
                    Console.WriteLine(linea);
                } while (linea != null);
            }
            Console.ReadLine();
        }
    }
}

```

(8.3.3) Prepara un programa que pregunte un nombre de fichero y muestre en pantalla el contenido de ese fichero, haciendo una pausa después de cada 24 líneas, para que dé tiempo a leerlo. Cuando el usuario pulse la tecla Intro, se mostrarán las siguientes 24 líneas, y así sucesivamente hasta que termine el fichero.

```

using System;
using System.IO;

namespace _8._3._3

```

```

{
    class Program
    {
        static void Main()
        {
            Console.Write("¿Qué fichero quieres consultar?: ");
            string nombreFichero = Console.ReadLine();

            StreamReader fichero = new
StreamReader("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
            "/EjerciciosTema8(Ficheros)/bin/Debug/" + nombreFichero);
            string linea;

            do
            {
                for (int i = 0; i < 24; i++)
                {
                    linea = fichero.ReadLine();
                    Console.WriteLine(linea);
                    if (linea == null) break;
                }
                Console.ReadLine();
            } while (fichero.ReadLine() != null);
        }
    }
}

```

(8.3.6) Crea un programa que pida al usuario el nombre de un fichero de texto y una frase a buscar, y que muestre en pantalla todas las líneas de ese fichero que contengan esa frase. Cada frase se debe preceder del número de línea (la primera línea del fichero será la 1, la segunda será la 2, y así sucesivamente).

```

using System;
using System.IO;

namespace _8._3._6
{
    class Program
    {
        static void Main()
        {
            Console.Write("Introduce el nombre de un fichero: ");
            string nombreFichero = Console.ReadLine();
            Console.Write("Introduce un texto para buscar: ");
            string textoBuscar = Console.ReadLine();
            StreamReader fichero = new
StreamReader("C:/Users/Tone/source/repos/EjerciciosTema8(Ficheros)" +
            "/EjerciciosTema8(Ficheros)/bin/Debug/" + nombreFichero);

            string linea;
            int contador = 1;
            do
            {
                linea = fichero.ReadLine();

                if ((linea != null) && (linea.Contains(textoBuscar)))
                {
                    Console.WriteLine("Línea {0}: " + linea, contador);
                }
            }
        }
    }
}

```

```

        contador++;
    } while (linea != null);
    fichero.Close();
    Console.ReadLine();
}
}
}

```

(8.4.1) Un programa que pida al usuario que teclee frases, y las almacene en el fichero "log.txt", que puede existir anteriormente (y que no deberá borrarse, sino añadir al final de su contenido). Cada sesión acabará cuando el usuario pulse Intro sin teclear nada.

Primero un proyecto para crear un archivo

```

using System;
using System.IO;

namespace EjerciciosTema8
{
    class Program
    {
        static void Main()
        {
            StreamWriter fichero = new StreamWriter("log.txt");
            fichero.WriteLine("Primera línea.");
            fichero.WriteLine("Segunda línea.");
            fichero.Close();

            Console.ReadLine();
        }
    }
}

```

Ahora el ejercicio

```

using System;
using System.IO;

namespace EjerciciosTema8
{
    class Program
    {
        static void Main()
        {
            string frase;
            StreamWriter fichero = new
StreamWriter("D:/Users/profe3d/source/repos/" +
"EjerciciosTema8/EjerciciosTema8/bin/Debug/log.txt", true);

            do
            {
                Console.Write("Introduce una frase: ");
                frase = Console.ReadLine();
                fichero.WriteLine(frase);

            } while (frase != "");

            fichero.Close();
        }
    }
}

```

```

    }
}

```

(8.5.1) Crea un programa que pida al usuario pares de números enteros y escriba su suma (con el formato "20 + 3 = 23") en pantalla y en un fichero llamado "sumas.txt", que se encontrará en un subdirectorío llamado "resultados". Cada vez que se ejecute el programa, deberá añadir los nuevos resultados a continuación de los resultados de las ejecuciones anteriores.

```

using System;
using System.IO;

namespace _8._5._1
{
    class Program
    {
        static void Main()
        {
            Console.Write("Introduce un número: ");
            int n1 = Convert.ToInt32(Console.ReadLine());
            Console.Write("Introduce otro número: ");
            int n2 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("{0} + {1} = {2}", n1, n2, n1 + n2);

            StreamWriter fichero = new
            StreamWriter("D:/Users/profe3d/source/repos/" +
                "EjerciciosTema8/8.5.1/bin/Debug/resultados/sumas.txt", true);

            fichero.WriteLine("{0} + {1} = {2}", n1, n2, n1 + n2);
            fichero.Close();

            Console.ReadLine();
        }
    }
}

```

(8.6.2) Mejora el ejemplo 08_06a para que no use "while (true)", sino una variable booleana de control.

```

using System;
using System.IO;

public class EncuentraFichero
{
    public static void Main()
    {
        StreamReader fichero;
        string nombre;
        bool terminado = false;

        while (!terminado)
        {
            Console.Write("Nombre del fichero (\"fin\" para terminar): ");
            nombre = Console.ReadLine();
            if (nombre == "fin") terminado = true;
            if (File.Exists(nombre))
            {
                fichero = new StreamReader(nombre);
            }
        }
    }
}

```

```

        Console.WriteLine("Su primera linea es: {0}",
fichero.ReadLine());
        fichero.Close();
    }
    else
    {
        if (nombre!="fin")
            Console.WriteLine("No existe!");
    }
}
}
}

```

(8.7.1) Un programa que pida al usuario el nombre de un fichero de origen y el de un fichero de destino, y que vuelque al segundo fichero el contenido del primero, convertido a mayúsculas. Se debe controlar los posibles errores, como que el fichero de origen no exista, o que el fichero de destino no se pueda crear.

```

using System;
using System.IO;

namespace _8._7._1
{
    class Program
    {
        static void Main()
        {
            StreamReader fichero = new
StreamReader("D:/Users/profe3d/source/repos/" +
"EjerciciosTema8/EjerciciosTema8/bin/Debug/registroDeUsuario.txt");
            StreamWriter fichero2 = new
StreamWriter("D:/Users/profe3d/source/repos/" +
"EjerciciosTema8/EjerciciosTema8/bin/Debug/contenedor.txt");
            string linea;

            do
            {
                linea = fichero.ReadLine();
                fichero2.WriteLine(linea);
            } while (linea != null);
            fichero.Close();
            fichero2.Close();
        }
    }
}

```

(8.7.2) Un programa que pida al usuario un número, una operación (+, -, *, /) y un segundo número, y muestre el resultado de la correspondiente operación. Si se teclea un dato no numérico, el programa deberá mostrar un aviso y volver a pedirlo, en vez de interrumpir la ejecución.

```

using System;

namespace _8._7._2
{
    class Program
    {

```

```

static void Main()
{
    do
    {
        try
        {
            Console.Write("Introduce un número (0 para salir): ");
            double n1 = Convert.ToDouble(Console.ReadLine());
            if (n1 == 0) break;
            Console.Write("Introduce una operación(+, -, *, /): ");
            char operacion = Convert.ToChar (Console.ReadLine());
            Console.Write("Introduce otro número: ");
            double n2 = Convert.ToDouble(Console.ReadLine());

            switch (operacion)
            {
                case '+': Console.WriteLine("{0} + {1} = {2}", n1, n2, n1
+ n2);
                    break;
                case '-': Console.WriteLine("{0} - {1} = {2}", n1, n2, n1
- n2);
                    break;
                case '*': Console.WriteLine("{0} * {1} = {2}", n1, n2, n1
* n2);
                    break;
                case '/': Console.WriteLine("{0} / {1} = {2}", n1, n2, n1
/ n2);
                    break;
                default: Console.WriteLine("Operación no válida.");
                    break;
            }
        }
        catch (Exception e)
        {
            Console.WriteLine("Ha habido un error: {0}", e.Message);
        }
    } while (true);
}
}

```

(8.7.3) Un programa que pida al usuario repetidamente pares de números y la operación a realizar con ellos (+, -, *, /) y guarde en un fichero "calculadora.txt" el resultado de dichos cálculos (con la forma "15 * 6 = 90"). Debe controlar los posibles errores, como que los datos no sean numéricos, la división entre cero, o que el fichero no se haya podido crear.

```

using System;
using System.IO;

namespace _8._7._3
{
    class Program
    {
        static void Main()
        {
            StreamWriter fichero = new
StreamWriter("E:/EjerciciosTema8/8.7.3/bin/Debug/calculadora.txt");
            do
            {
                try

```



```

    {
        Console.WriteLine("Introduce un número (0 para salir): ");
        double n1 = Convert.ToDouble(Console.ReadLine());
        if (n1 == 0) break;
        Console.WriteLine("Introduce una operación(+, -, *, /): ");
        char operacion = Convert.ToChar(Console.ReadLine());
        Console.WriteLine("Introduce otro número: ");
        double n2 = Convert.ToDouble(Console.ReadLine());

        switch (operacion)
        {
            case '+':
                fichero.WriteLine("{0} + {1} = {2}", n1, n2, n1 +
n2);
                break;
            case '-':
                fichero.WriteLine("{0} - {1} = {2}", n1, n2, n1 -
n2);
                break;
            case '*':
                fichero.WriteLine("{0} * {1} = {2}", n1, n2, n1 *
n2);
                break;
            case '/':
                fichero.WriteLine("{0} / {1} = {2}", n1, n2, n1 /
n2);
                break;
            default:
                Console.WriteLine("Operación no válida.");
                break;
        }
    }
    catch (DivideByZeroException e) //No está funcionando esta
excepción
    {
        Console.WriteLine("Ha habido un error: {0}", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Ha habido un error: {0}", e.Message);
    }
} while (true);
fichero.Close();
}
}
}

```

(8.9.1) Abre un fichero con extensión EXE (cuyo nombre introducirá el usuario) y comprueba si realmente se trata de un ejecutable, mirando si los dos primeros bytes del fichero son un 77 (que corresponde a una letra "M", según el estándar que marca el código ASCII) y un 90 (una letra "Z"), respectivamente.

```

using System;
using System.IO;

public class Program
{
    public static void Main()
    {

```

```

Console.Write("Introduce un nombre de archivo .exe: ");
string nombre = Console.ReadLine();
try
{
    FileStream fichero = new FileStream(nombre, FileMode.Open);
    for (int i = 0; i < 2; i++)
    {
        byte unDato = (byte)fichero.ReadByte();
        Console.WriteLine("El {0}º byte leído es un {1}", i+1, unDato);
    }
    fichero.Close();
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error: " + e.Message);
    Console.ReadLine();
    return;
}
}
}

```

(8.9.2) Abre una imagen BMP (cuyo nombre introducirá el usuario) y comprueba si realmente se trata de un fichero en ese formato, mirando si los dos primeros bytes del fichero corresponden a una letra "B" y una letra "M", respectivamente.

```

using System;
using System.IO;

public class Program
{
    public static void Main()
    {
        Console.Write("Introduce un nombre de archivo .bmp: ");
        string nombre = Console.ReadLine();
        try
        {
            FileStream fichero = new FileStream(nombre, FileMode.Open);
            for (int i = 0; i < 2; i++)
            {
                byte unDato = (byte)fichero.ReadByte();
                Console.WriteLine("El {0}º byte leído es un {1}", i+1, unDato);
            }
            fichero.Close();
            Console.ReadLine();
        }
        catch (Exception e)
        {
            Console.WriteLine("Error: " + e.Message);
            Console.ReadLine();
            return;
        }
    }
}

```

(8.9.3) Abre una imagen GIF y comprueba si sus tres primeros bytes son las letras G, I, F.

```

using System;

```

```

using System.IO;

public class Program
{
    public static void Main()
    {
        Console.Write("Introduce un nombre de archivo .gif: ");
        string nombre = Console.ReadLine();
        try
        {
            FileStream fichero = new FileStream(nombre, FileMode.Open);
            for (int i = 0; i < 3; i++)
            {
                byte unDato = (byte)fichero.ReadByte();
                Console.WriteLine("El {0}º byte leído es un {1}", i+1, unDato);
            }
            fichero.Close();
            Console.ReadLine();
        }
        catch (Exception e)
        {
            Console.WriteLine("Error: " + e.Message);
            Console.ReadLine();
            return;
        }
    }
}

```

(8.10.1) Crea un programa que compruebe si un fichero (cuyo nombre introducirá el usuario) contiene una cierta letra (también escogida por el usuario).

```

using System;
using System.IO;

namespace _8._10._1
{
    class Program
    {
        static void Main()
        {
            Console.Write("Introduce un nombre de fichero: ");
            string nombre = Console.ReadLine();
            Console.Write("Introduce la letra que quieres buscar: ");
            char letra = Convert.ToChar(Console.ReadLine());

            FileStream fichero = new FileStream(nombre, FileMode.Open);
            for (int i=0; i<fichero.Length; i++)
            {
                char dato = Convert.ToChar (fichero.ReadByte());
                if (letra == dato)
                {
                    Console.WriteLine("Sí, la letra {0} aparece en la línea {1}",
letra, i + 1);
                }
            }
            Console.ReadLine();
        }
    }
}

```

(8.10.2) Crea un programa que cuente la cantidad de vocales que contiene un fichero binario.

```
using System;
using System.IO;

namespace _8._10._2
{
    class Program
    {
        static void Main()
        {
            int contador = 0;

            Console.Write("Introduce un nombre de fichero: ");
            string nombre = Console.ReadLine();

            FileStream fichero = new FileStream(nombre, FileMode.Open);
            for (int i=0; i<fichero.Length; i++)
            {
                char dato = Convert.ToChar (fichero.ReadByte());
                if (dato == 'a' || dato == 'e' || dato == 'i' || dato == 'o' ||
dato == 'u')
                {
                    contador++;
                }
            }
            Console.Write(contador);
            Console.ReadLine();
        }
    }
}
```

(8.10.3) Crea un programa que diga si un fichero (binario) contiene una cierta palabra que introduzca el usuario.

```
using System;
using System.IO;
using System.Text;

namespace _8._10._3
{
    class Program
    {
        static void Main()
        {
            Console.Write("Introduce un nombre de fichero: ");
            string nombre = Console.ReadLine();
            Console.Write("Introduce una palabra: ");
            string palabraUsuario = Console.ReadLine();

            byte[] bytesEnPalabraUsuario =
Encoding.ASCII.GetBytes(palabraUsuario);

            FileStream fichero = new FileStream(nombre, FileMode.Open);
            bool encontrada = false;
            for (int i=0; i<fichero.Length; i++)
            {
                byte dato = (byte)fichero.ReadByte();
                int j;
                for (j=0; j<bytesEnPalabraUsuario.Length; j++)
                {
```

```

        if (dato == bytesEnPalabraUsuario[j])
        {
            dato = (byte)fichero.ReadByte();
        }
        else break;
    }

    if (j == bytesEnPalabraUsuario.Length)
    {
        encontrada = true;
        break;
    }
}
if (encontrada)
{
    Console.WriteLine("Se ha encontrado la palabra en el fichero.");
}
else Console.WriteLine("No se ha encontrado la palabra.");
Console.ReadLine();
}
}
}

```

(8.10.4) Crea un programa que extraiga a un fichero de texto todos los caracteres alfabéticos (códigos 32 a 127, además del 10 y el 13) que contenga un fichero binario.

```

using System;
using System.IO;

namespace _8._10._4
{
    class Program
    {
        static void Main()
        {
            StreamWriter writer = new StreamWriter("SinSignos.txt", false);
            Console.Write("Introduce un nombre de fichero: ");
            string nombre = Console.ReadLine();

            FileStream fichero = new FileStream(nombre, FileMode.Open);
            for (int i=0; i<fichero.Length; i++)
            {
                //Si queremos que se muestre su número ascii
                //byte dato = (byte) fichero.ReadByte();
                char dato = Convert.ToChar(fichero.ReadByte());
                if (dato>=32 && dato<=127)
                {
                    writer.Write(dato);
                }
            }
            writer.Close();
            Console.ReadLine();
        }
    }
}

```

(8.11.1) Abre un fichero con extensión EXE y comprobar si realmente se trata de un ejecutable, mirando si los dos primeros bytes del fichero corresponden a una letra "M" y una letra "Z", respectivamente. Debes leer ambos bytes a la vez, usando un array.

```

using System;

```

```

using System.IO;

public class Program
{
    public static void Main()
    {
        Console.Write("Introduce un nombre de archivo .exe: ");
        string nombre = Console.ReadLine();
        try
        {
            FileStream fichero = new FileStream(nombre, FileMode.Open);

            byte[] datos = new byte[2];
            int posicion = 0;
            int cantidadALeer = 2;
            fichero.Read(datos, posicion, cantidadALeer);

            for (int i=0; i<datos.Length; i++)
            {
                Console.WriteLine("El {0}º dato del array leído es un {1}", i+1,
datos[i]);
            }
            fichero.Close();
            Console.ReadLine();
        }
        catch (Exception e)
        {
            Console.WriteLine("Error: " + e.Message);
            Console.ReadLine();
            return;
        }
    }
}

```

(8.11.2) Abre una imagen BMP (cuyo nombre introducirá el usuario) y comprueba si realmente se trata de un fichero en ese formato, mirando si los dos primeros bytes del fichero corresponden a una letra "B" y una letra "M", respectivamente. Usa "Read" y un array.

```

using System;
using System.IO;

public class Program
{
    public static void Main()
    {
        Console.Write("Introduce un nombre de archivo .bmp: ");
        string nombre = Console.ReadLine();
        try
        {
            FileStream fichero = new FileStream(nombre, FileMode.Open);

            byte[] datos = new byte[2];
            int posicion = 0;
            int cantidadALeer = 2;
            fichero.Read(datos, posicion, cantidadALeer);

            for (int i=0; i<datos.Length; i++)
            {
                Console.WriteLine("El {0}º dato del array leído es un {1}", i+1,
datos[i]);
            }
        }
    }
}

```

```

    }
    fichero.Close();
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error: " + e.Message);
    Console.ReadLine();
    return;
}
}
}

```

(8.11.3) Abre una imagen GIF y comprueba si sus tres primeros bytes son las letras G, I, F. Guarda los 3 datos en un array.

```

using System;
using System.IO;

public class Program
{
    public static void Main()
    {
        Console.Write("Introduce un nombre de archivo .gif: ");
        string nombre = Console.ReadLine();
        try
        {
            FileStream fichero = new FileStream(nombre, FileMode.Open);

            byte[] datos = new byte[3];
            int posicion = 0;
            int cantidadALeer = 3;
            fichero.Read(datos, posicion, cantidadALeer);

            for (int i=0; i<datos.Length; i++)
            {
                Console.WriteLine("El {0}º dato del array leído es una {1}", i+1,
Convert.ToChar (datos[i]));
            }
            fichero.Close();
            Console.ReadLine();
        }
        catch (Exception e)
        {
            Console.WriteLine("Error: " + e.Message);
            Console.ReadLine();
            return;
        }
    }
}

```

(8.11.4) Abre una imagen en formato BMP y comprueba si está comprimida, mirando el valor del byte en la posición 30 (empezando a contar desde 0). Si ese valor es un 0 (que es lo habitual), indicará que el fichero no está comprimido. Deberás leer toda la cabecera (los primeros 54 bytes) con una sola orden.

```

using System;
using System.IO;

```

```

public class Program
{
    public static void Main()
    {
        Console.WriteLine("Introduce un nombre de archivo .bmp: ");
        string nombre = Console.ReadLine();
        try
        {
            FileStream fichero = new FileStream(nombre, FileMode.Open);

            byte[] datos = new byte[54];
            int posicion = 0;
            int cantidadALeer = 54;
            fichero.Read(datos, posicion, cantidadALeer);

            for (int i=0; i<datos.Length; i++)
            {
                Console.WriteLine("El {0}º dato del array leído es un {1}", i+1,
datos[i]);
            }
            if (datos[30] == 0) Console.WriteLine("\nEl archivo no está comprimido");
            else Console.WriteLine("\nEl archivo está comprimido");
            fichero.Close();
            Console.ReadLine();
        }
        catch (Exception e)
        {
            Console.WriteLine("Error: " + e.Message);
            Console.ReadLine();
            return;
        }
    }
}

```

(8.12.1) Abre un fichero con extensión EXE y comprueba si su segundo byte corresponde a una letra "Z", sin leer su primer byte.

```

using System;
using System.IO;

namespace _8._12._1
{
    class Program
    {
        static void Main()
        {
            FileStream fichero = new FileStream("prueba.exe", FileMode.Open);
            fichero.Seek(1, SeekOrigin.Begin);
            //int dato = fichero.ReadByte();
            //Si queremos un número ascii
            char dato = Convert.ToChar(fichero.ReadByte());
            Console.WriteLine(dato);
            Console.ReadLine();
        }
    }
}

```


(8.12.2) Abre una imagen en formato BMP y comprueba si está comprimida, mirando el valor del byte en la posición 30 (empezando a contar desde 0). Si ese valor es 0 (que es lo habitual), indicará que el fichero no está comprimido. Salta a esa posición directamente, sin leer toda la cabecera.

```
using System;
using System.IO;

namespace _8._12._1
{
    class Program
    {
        static void Main()
        {
            FileStream fichero = new FileStream("verisign.bmp", FileMode.Open);
            fichero.Seek(30, SeekOrigin.Begin);
            int dato = fichero.ReadByte();
            if (dato == 0) Console.WriteLine("El dato en la posición 30 es {0}, es
decir, no está comprimida la imagen.", dato);
            else Console.WriteLine("La imagen está comprimida. El dato es: {0}",
dato);
            Console.ReadLine();
        }
    }
}
```

(8.13.1) Abre un fichero con extensión EXE y comprueba si comienza con el entero corto 23117.

```
using System;
using System.IO;

namespace _8._13._1
{
    class Program
    {
        static void Main()
        {
            BinaryReader fichero = new BinaryReader(File.Open("prueba.exe",
FileMode.Open));
            short dato = fichero.ReadInt16();
            fichero.Close();
            Console.WriteLine(dato);
            Console.ReadLine();
        }
    }
}
```

(8.13.2) Abre una imagen en formato BMP y comprueba si comienza con el entero corto 19778.

```
using System;
using System.IO;

namespace _8._13._1
{
    class Program
    {
        static void Main()
        {
```

```

        BinaryReader fichero = new BinaryReader(File.Open("verisign.bmp",
 FileMode.Open));
        short dato = fichero.ReadInt16();
        fichero.Close();
        Console.Write(dato);
        Console.ReadLine();
    }
}

```

(8.13.3) El alto de un fichero BMP es un entero de 32 bits que se encuentra en la posición 22. Amplía el ejemplo 08_13d, para que muestre también el alto de ese fichero BMP.

```

using System;
using System.IO;

namespace _8._13._3
{
    class Program
    {
        static void Main()
        {
            BinaryReader fichero = new BinaryReader(File.Open("verisign.bmp",
 FileMode.Open));
            fichero.BaseStream.Seek(18, SeekOrigin.Begin);
            int ancho = fichero.ReadInt32();
            fichero.BaseStream.Seek(22, SeekOrigin.Begin);
            int alto = fichero.ReadInt32();
            fichero.Close();
            Console.WriteLine("El alto de la imagen es: {0} y el ancho: {1}", alto,
 ancho);
            Console.ReadLine();
        }
    }
}

```

(8.15.1) Crea una copia de un fichero EXE. La copia debe tener el mismo nombre y extensión BAK.

```

using System;
using System.IO;

namespace _8._15._1
{
    class Program
    {
        static void Main()
        {
            BinaryReader ficheroOriginal; //Mejor hacerlo con FileStream
            BinaryWriter copia;

            try
            {
                ficheroOriginal = new BinaryReader(File.Open("nombre.exe",
 FileMode.Open));
                copia = new BinaryWriter(File.Open("nombre.bak",
 FileMode.Create));
                for (int i=0; i<ficheroOriginal.BaseStream.Length; i++)
                {
                    copia.Write(ficheroOriginal.ReadByte());
                }
            }
            catch { }
        }
    }
}

```

```

        }

        ficheroOriginal.Close();
        copia.Close();
    }

    catch (Exception error)
    {
        Console.WriteLine("Ha ocurrido un error al leer el fichero: {0}",
error.Message);
        return;
    }
    Console.WriteLine("Archivos copiados con éxito");
    Console.ReadLine();
}
}
}

```

(8.15.2) Crea un programa que "encripte" el contenido de un fichero BMP, volcando su contenido a un nuevo fichero, en el que intercambiará los dos primeros bytes. Para desencriptar, bastará con volver a intercambiar esos dos bytes, volcando a un tercer fichero.

```

using System;
using System.IO;

namespace _8._15._2
{
    class Program
    {
        static void Main()
        {
            try
            {
                CopiarFicheroConLosDosPrimerosBytesDados("verisign.bmp",
"encriptado.bmp", Convert.ToByte('M'), Convert.ToByte('B'));
                Console.WriteLine("Encriptación realizada con éxito.");
                Console.ReadLine();
                CopiarFicheroConLosDosPrimerosBytesDados("encriptado.bmp",
"imagenFinal.bmp", Convert.ToByte('B'), Convert.ToByte('M'));
                Console.WriteLine("Desencriptación realizada con éxito.");
            }
            catch (Exception error)
            {
                Console.WriteLine("Ha habido un error en la encriptación: {0}",
error.Message);
            }
            Console.ReadLine();
        }

        private static void CopiarFicheroConLosDosPrimerosBytesDados(string
nombreFicheroOriginal, string nombreFicheroDestino, byte byte1, byte byte2)
        {
            FileStream ficheroOriginal = File.OpenRead(nombreFicheroOriginal);
            FileStream ficheroEncriptado = File.Create(nombreFicheroDestino);
            ficheroEncriptado.WriteByte(byte1);
            ficheroEncriptado.WriteByte(byte2);
            ficheroOriginal.Seek(2, SeekOrigin.Begin);
            int dato;
            do
            {
                dato = ficheroOriginal.ReadByte();
                if (dato != -1)

```

```

        {
            ficheroEncriptado.WriteByte((byte)dato);
        }
    } while (dato != -1);

    ficheroEncriptado.Close();
    ficheroOriginal.Close();
}
}
}

```

(8.16.1) Crea un programa que "encripte" el contenido de un fichero BMP, intercambiando los dos primeros bytes (y modificando el mismo fichero). Para desencriptar, bastará con volver a intercambiar esos dos bytes.

```

using System;
using System.IO;

namespace _8._16._1
{
    class Program
    {
        static void Main(string[] args)
        {
            try //Hacerlo también con FileStream
            {
                BinaryReader fichero = new BinaryReader(File.Open("verisign.bmp",
                FileMode.Open, FileAccess.ReadWrite));
                byte dato0 = fichero.ReadByte();
                byte dato1 = fichero.ReadByte();
                fichero.BaseStream.Seek(0, SeekOrigin.Begin);
                fichero.BaseStream.WriteByte(dato1);
                fichero.BaseStream.Seek(1, SeekOrigin.Begin); //Esta línea no
                hace falta porque ya estamos en esta posición
                fichero.BaseStream.WriteByte(dato0);

                fichero.Close();
            }
            catch (Exception error)
            {
                Console.WriteLine("Ha habido un error: {0}", error.Message);
                return;
            }
            Console.ReadLine();
        }
    }
}

```

(8.16.2) Crea un programa que vuelque todo el contenido de un fichero de texto a otro, convirtiendo cada frase a mayúsculas. Los nombres de ambos ficheros se deben indicar como parámetros en la línea de comandos.

```

using System;
using System.IO;

namespace _8._16._2
{
    class Program
    {
        static void Main(string[] args)
        {
            if (args.Length != 2)

```

```

        {
            Console.WriteLine("Numero de argumentos inválido. Es necesario
especificar los " +
                "nombres de los archivos de origen y destino.");
            return;
        }

        try
        {
            StreamReader ficheroOriginal = new StreamReader(args[0]);
            StreamWriter ficheroFinal = new StreamWriter(args[1]);
            string linea;
            do
            {
                linea = ficheroOriginal.ReadLine();
                if (linea != null)
                {
                    ficheroFinal.WriteLine(linea.ToUpper());
                }
            } while (linea != null);
            ficheroOriginal.Close();
            ficheroFinal.Close();
        }
        catch (Exception error)
        {
            Console.WriteLine("Ha habido un error: {0}", error.Message);
            return;
        }
    }
}

```

(8.16.3) Un programa que vuelque todo el contenido de un fichero binario a otro, convirtiendo cada letra entre la A y la Z a mayúsculas. Los nombres de ambos ficheros se deben indicar como parámetros en la línea de comandos. Si no hay parámetros, se le preguntará al usuario.

```

using System;
using System.IO;

namespace _8._16._3
{
    class Program
    {
        static void Main(string [] args)
        { //Con el editor notePad ++ (gratis) se puede abrir el fichero y
comprobar si el programa hace lo que debe
            try
            {
                BinaryReader ficheroOriginal = new
BinaryReader(File.Open("verisign.bmp", FileMode.Open));
                BinaryWriter ficheroFinal = new
BinaryWriter(File.Open("ficheroFinal.bmp", FileMode.Create));
                for (int i = 0; i < ficheroOriginal.BaseStream.Length; i++)
                //Para ahorrar recursos, sería mejor utilizar un while para no tener que leer la
longitud del fichero cada vez que leemos
                {
                    byte dato = ficheroOriginal.ReadByte();
                    if (dato >=97 && dato <= 122)
                    {
                        ficheroFinal.Write(dato - 32);
                    }
                }
            }
        }
    }
}

```

```

        else
            ficheroFinal.Write(dato);
    }
    ficheroOriginal.Close();
    ficheroFinal.Close();
}
catch (Exception error)
{
    Console.WriteLine("Ha habido un error: {0}", error.Message);
}
}
}
}

```

(8.16.5) Crea un "traductor básico de C# a C", que volcará todo el contenido de un fichero de texto a otro, pero reemplazando "Console.WriteLine" con "printf", "Main" con "main", "string" con "char[80]", "Console.ReadLine" con "scanf", y eliminando "static" y "public" y las líneas que comiencen con "Using".

```

using System;
using System.IO;

namespace _8._16._5
{
    class Program
    {
        static void Main()
        {
            StreamReader ficheroOriginal = new
StreamReader("documentoOriginal.txt");
            StreamWriter ficheroFinal = new StreamWriter("documentoFinal.txt");
            string linea;
            do
            {
                linea = ficheroOriginal.ReadLine();
                if (linea != null)
                {
                    linea.Replace("Console.WriteLine", "printf");
                    linea.Replace("Main", "main");
                    linea.Replace("string", "char[80]");
                    linea.Replace("static", "");
                    linea.Replace("public", "");

                    if (!linea.StartsWith("using"))
                        ficheroFinal.WriteLine(linea);
                }
            } while (linea != null);

            ficheroOriginal.Close();
            ficheroFinal.Close();
        }
    }
}

```

(8.16.9) Crea un programa que compare si dos ficheros son iguales byte a byte, comprobando primero su tamaño y leyendo después cada vez un byte de cada uno de ellos.

```

using System;
using System.IO;

```

```

namespace _8._16._9
{
    class Program
    {
        static void Main()
        {
            BinaryReader fichero = new BinaryReader(File.Open("original.bmp",
            FileMode.Open));
            BinaryReader fichero2 = new BinaryReader(File.Open("copia.bmp",
            FileMode.Open));
            bool iguales = true;

            if (fichero.BaseStream.Length == fichero2.BaseStream.Length)
            {
                Console.WriteLine("El tamaño de la imagen es el mismo: {0} y
            {1}", fichero.BaseStream.Length, fichero2.BaseStream.Length);
            }
            else Console.WriteLine("El tamaño de la imagen es distinto: {0} y
            {1}", fichero.BaseStream.Length, fichero2.BaseStream.Length);

            for (int i = 0; i < fichero.BaseStream.Length; i++)
            {
                if (fichero.ReadByte() != fichero2.ReadByte())
                {
                    iguales = false;
                    break;
                }
            }

            if (iguales)
                Console.WriteLine("Los ficheros son iguales.");
            else
                Console.WriteLine("Los ficheros son distintos.");

            fichero.Close();
            fichero2.Close();
            Console.ReadLine();
        }
    }
}

```

(8.16.10) Crea un programa que compare si dos ficheros son iguales, volcando todo su contenido en sendos arrays, que compararás.

```

using System;
using System.IO;

namespace _8._16._10
{
    class Program
    {
        static void Main()
        {
            BinaryReader fichero = new BinaryReader(File.Open("original.bmp",
            FileMode.Open));
            BinaryReader fichero2 = new BinaryReader(File.Open("image1.gif",
            FileMode.Open));
            byte[] datos = new byte[fichero.BaseStream.Length]; //Es innecesario
            volcar los contenidos en arrays.
            //De hecho, es mejor no hacerlo para ahorrar recursos

```

```

byte[] datos2 = new byte[fichero2.BaseStream.Length];
bool comparacion = true;

if (fichero.BaseStream.Length == fichero2.BaseStream.Length)
{
    for (int i = 0; i < fichero.BaseStream.Length; i++)
    {
        datos[i] = fichero.ReadByte();
        datos2[i] = fichero2.ReadByte();
        if (datos[i] != datos2[i])
        {
            comparacion = false;
            break;
        }
    }
}
else
    comparacion = false;

if (comparacion)
{
    Console.WriteLine("Los ficheros son iguales.");
}
else Console.WriteLine("Los ficheros son distintos.");

fichero.Close();
fichero2.Close();
Console.ReadLine();
}
}
}

```

(8.16.11) Un programa que muestre el nombre del autor de un fichero de música en formato MP3 (tendrás que localizar en Internet la información sobre dicho formato y sobre la cabecera ID3 V1, que se encuentra -si está presente- en los últimos 128 bytes del fichero).

```

using System;
using System.IO;

namespace _8._16._11
{
    class Program
    {
        static void Main()
        {
            byte[] bytesDeAutor = new byte[30];
            FileStream fichero = File.OpenRead("musica.mp3");
            fichero.Seek(98, SeekOrigin.End);
            fichero.Read(bytesDeAutor, 0, 30);
            Console.WriteLine("El autor es: " +
                System.Text.Encoding.ASCII.GetString(bytesDeAutor));
            Console.ReadLine();
        }
    }
}

```