

## Ejercicios de ficheros

1. Escribir números en un fichero binario.
2. Leer números de un fichero binario.
3. Contar vocales de un fichero binario.
4. Sumar números de un fichero de texto.
5. Escribir y leer un fichero de texto.
6. Contar vocales de un fichero de texto.
7. Listar el contenido de un directorio.
8. Filtrar ficheros.
9. Crear directorios.
10. Clase persona serializable
11. Escribir en un fichero serializable
12. Leer de un fichero serializable objetos personas.

### 1.- Escribir números en un fichero binario

Lee números enteros por teclado y los escribe en el fichero datos.dat. La lectura de datos acaba cuando se introduce -1.

```
import java.io.DataOutputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;

public class escribirFicheroBinario {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        FileOutputStream fos = null;
        DataOutputStream salida = null;
        int n;

        try {
            fos = new FileOutputStream("datos.dat");
            salida = new DataOutputStream(fos);

            System.out.print("Introduce número entero. -1 para acabar: ");
            n = sc.nextInt();
            while (n != -1) {
                salida.writeInt(n); //se escribe el número entero en el fichero
                System.out.print("Introduce número entero. -1 para acabar: ");
                n = sc.nextInt();
            }
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (IOException e) {
            System.out.println(e.getMessage());
        } finally {
            try {
                if (fos != null) {
                    fos.close();
                }
                if (salida != null) {
                    salida.close();
                }
            } catch (IOException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
```

## 2.- Leer números de un fichero binario

Lee números del fichero datos.dat. Utilizamos un bucle cuando se llega al final del fichero se lanza la excepción EOFException para salir del bucle while.

```
import java.io.DataInputStream;
import java.io.EOFException;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class leerFicheroBinario {

    public static void main(String[] args) {
        FileInputStream fis = null;
        DataInputStream entrada = null;
        int n;
        try {
            fis = new FileInputStream("datos.dat");
            entrada = new DataInputStream(fis);
            while (true) {
                n = entrada.readInt(); //se lee un entero del fichero
                System.out.println(n); //se muestra en pantalla
            }
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (EOFException e) {
            System.out.println("Fin de fichero");
        } catch (IOException e) {
            System.out.println(e.getMessage());
        } finally {
            try {
                if (fis != null) {
                    fis.close();
                }
                if (entrada != null) {
                    entrada.close();
                }
            } catch (IOException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
```

## 3.- Contar vocales de un fichero binario

```
import java.io.*;

public class contarVocalesFicheroBinario {

    public static void main( String[] args )
    {
        // Cantidad de "a" en un fichero de cualquier tipo
        // Mirando errores solo con try-catch

        System.out.println("Contando vocales...");
        int a,e,i,o,u;
        a=e=i=o=u= 0;

        try
        {
            FileInputStream ficheroEntrada2 =
                new FileInputStream(new File("vocales.dat"));
            int dato;
```

```

        while ((dato = ficheroEntrada2.read()) != -1) {
            switch(dato) {
                case 97:      a++;      // Código ASCII de "a"
                            break;
                case 101:     e++;      // Código ASCII de "e"
                            break;
                case 105:     i++;      // Código ASCII de "i"
                            break;
                case 111:     o++;      // Código ASCII de "o"
                            break;
                case 117:     u++;      // Código ASCII de "u"
                            break;
                default:      break;
            }

            ficheroEntrada2.close();
        }
        catch (Exception errorDeFichero)
        {
            System.out.println(
                "Ha habido problemas: " +
                errorDeFichero.getMessage() );
        }

        System.out.println("Cantidad de \"a\": " + a);
        System.out.println("Cantidad de \"e\": " + e);
        System.out.println("Cantidad de \"i\": " + i);
        System.out.println("Cantidad de \"o\": " + o);
        System.out.println("Cantidad de \"u\": " + u);
    }
}

```

#### 4.- Sumar números de un fichero

Escribe un programa que reciba el nombre de un archivo que almacena una serie de números enteros positivos en distintas líneas e imprima su suma. [giltesa.com](http://giltesa.com)

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class sumarNumerosFichero
{
    public static void main(String[] args) {

        System.out.println("Nombre del fichero que contiene los numeros:");
        Scanner teclado = new Scanner(System.in);
        try
        {
            FileReader fr = new FileReader(teclado.nextLine());
            BufferedReader br = new BufferedReader(fr);
            int num = 0;
            String temp;
            while((temp = br.readLine())!=null)
            {
                num+= Integer.parseInt(temp);
                System.out.println(temp);
            }
            fr.close();
            System.out.println("Los numeros suman: " + num);
        }
        catch( IOException | NumberFormatException e )
        { System.out.println(e.getMessage()); }

    }
}

```

**5.- Escribir y leer un fichero de texto.**

Crea un programa que escriba en un fichero de texto y lea el contenido de este. ([discoduroderoer.es](http://discoduroderoer.es))

```
import java.io.*;
import java.util.Scanner;

public class leerescribirFichero {
    public static void main(String[] args) {

        String nombre_fichero;
        try{
            //Pedimos el nombre del fichero
            System.out.print("Nombre del fichero a mostrar:");
            Scanner teclado = new Scanner(System.in);
            nombre_fichero = teclado.nextLine();

            BufferedReader br=new BufferedReader(new FileReader(nombre_fichero));
            BufferedWriter bw=new BufferedWriter(new FileWriter(nombre_fichero));

            escribeFichero(bw);
            bw.flush();
            leeFichero(br);
        }catch(IOException e){
            System.out.println("Error E/S: "+e);
        }
    }

    public static void escribeFichero(BufferedWriter bw) throws IOException{
        System.out.println("Escribiendo...");
        bw.write("Esto es una prueba usando Buffered");
        bw.newLine();
        bw.write("Seguimos usando Buffered");
    }

    public static void leeFichero(BufferedReader br) throws IOException{
        System.out.println("Leyendo.....");
        String linea=br.readLine();
        while(linea!=null){
            System.out.println(linea);
            linea=br.readLine();
        }
    }
}
```

**6.- Contar vocales de un fichero de texto.**

Dado un fichero de texto contar el número de vocales que hay en él. ([lawebdelprogramador.com](http://lawebdelprogramador.com))

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class contarVocales {

    public static void main(String[] args) {

        String nombre_fichero;
        int a, e, i, o, u;
        a = e = i = o = u = 0;

        try {
            //Pedimos el nombre del fichero
```

```

System.out.print("Nombre del fichero a mostrar:");
Scanner teclado = new Scanner(System.in);
nombre_fichero = teclado.nextLine();

BufferedReader br=new BufferedReader(new FileReader(nombre_fichero));
String linea=br.readLine();
while(linea!=null){
    System.out.println(linea);
    int longitud = linea.length();
    int indice=0;
    char letra;
    do {
        letra = linea.toLowerCase().charAt(indice);
        switch(letra) {
            case 'a':    a++; break;
            case 'e':    e++; break;
            case 'i':    i++; break;
            case 'o':    o++; break;
            case 'u':    u++; break;
            default:     break;
        }
        indice++;
    }while(indice < longitud);
    linea=br.readLine();
}

System.out.println("Cantidad de letras 'a' en la frase: " + a);
System.out.println("Cantidad de letras 'e' en la frase: " + e);
System.out.println("Cantidad de letras 'i' en la frase: " + i);
System.out.println("Cantidad de letras 'o' en la frase: " + o);
System.out.println("Cantidad de letras 'u' en la frase: " + u);

} catch (FileNotFoundException ex) {
    System.out.println("Error ficheros: "+ex);
} catch (IOException ex) {
    System.out.println("Error lectura: "+ex);
}
}
}

```

## 7.- Listar un directorio:

Escribir un programa que liste los nombres de los archivos y directorios que se encuentren en el directorio que se pasa. [giltesa.com](http://giltesa.com)

```

import java.io.File;
import java.util.Scanner;

public class listarDirectorio
{
    public static void main(String[] args)
    {
        String ruta; //"C:\\Users\\profesor\\Desktop\\ejemplosUD6";

        //Pedimos el nombre del fichero

        System.out.print("Escribe la ruta:");
        Scanner teclado = new Scanner(System.in);
        ruta = teclado.nextLine();
        File file = new File(ruta);
        if( file.isDirectory() )
        {
            File[] ficheros = file.listFiles();
            for( File f : ficheros )
                System.out.println(f.getName());
        }
    }
}

```

## 8.- FiltrarFicheros

Escribir un programa que muestre los nombres de los archivos de un directorio, donde la extensión coincida con la que se pasa. ([giltesa.com](http://giltesa.com))

```
import java.io.File;
import java.util.Scanner;

public class filtrarFicheros{

    public static void main (String[] args){

        String nombre_fichero, extension;
        //"C:\\Users\\profesor\\Desktop\\ejemplosUD6";
        //Pedimos el nombre del fichero
        System.out.print("Escribe la ruta:");
        Scanner teclado = new Scanner(System.in);
        nombre_fichero = teclado.nextLine();

        System.out.print("Escribe la extension:");
        extension = teclado.nextLine();

        File file = new File(nombre_fichero);
        if( file.isDirectory() )
        {
            String[] ficheros = file.list();
            for( String f : ficheros )
            { if( f.endsWith(extension) )
              { System.out.println(f);}
            }
        }
    }
}
```

## 9.- Crear directorios

Escribir un programa que cree directorios o carpetas en una ruta especificada.

```
import java.io.File;
public class crearDirectorios {

    public static void main(String[] args) {
        try {
            // Declaración de variables
            String directorio = "C:\\Users\\profesor\\Desktop\\ejemplosUD6\\";
            String varios = "carpetal/carpet2/carpet3";

            // Crear un directorio
            boolean exito = (new File(directorio)).mkdir();
            if (exito)
                System.out.println("Directorio: " + directorio + " creado");
            // Crear varios directorios
            exito = (new File(varios)).mkdirs();
            if (exito)
                System.out.println("Directorios: " + varios + " creados");
        } catch (Exception e){
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

**10.- Clase persona serializable**

Clase persona, con los atributos id de tipo entero y nombre, apellido1 y apellido2 de tipo String

```
import java.io.Serializable;

public class Persona implements Serializable {
    private int id;
    private String nombre;
    private String apellido1, apellido2;

    public Persona(int id, String nombre, String apellido1, String apellido2) {
        this.id = id;
        this.nombre = nombre;
        this.apellido1 = apellido1;
        this.apellido2 = apellido2;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellido1() {
        return apellido1;
    }

    public void setApellido1(String apellido1) {
        this.apellido1 = apellido1;
    }

    public String getApellido2() {
        return apellido2;
    }

    public void setApellido2(String apellido2) {
        this.apellido2 = apellido2;
    }

    public String toString(){
        return    this.id+" -> "+this.nombre+" "+this.apellido1+" "+this.apellido2;
    }
}
```

**11.- Escribir en un fichero serializable**

Escribir en un fichero objetos personas.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

public class escribirSerializable {

    public static void main(String[] args) {
        Persona p;

        try {
            File fichero = new File("persona.dat");
            FileOutputStream fos = new FileOutputStream(fichero);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            p=new Persona(1,"Jesus","Vallejo","Lazaro");
            oos.writeObject(p);
            p=new Persona(2,"Borja","Mayoral","Moya");
            oos.writeObject(p);
            p=new Persona(3,"Oliver","Torres","Muñoz");
            oos.writeObject(p);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

**12.- Leer de un fichero serializable objetos personas.**

Lee de un fichero los objetos personas y lo muestra por pantalla.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.ObjectInputStream;

public class leerSerializable {

    public static void main(String[] args) {
        Persona p;

        try {
            File fichero = new File("persona.dat");
            FileInputStream fis = new FileInputStream(fichero);
            ObjectInputStream ois = new ObjectInputStream(fis);
            p=(Persona)ois.readObject();
            System.out.println(p.toString());
            p=(Persona)ois.readObject();
            System.out.println(p.toString());
            p=(Persona)ois.readObject();
            System.out.println(p.toString());
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```