

Ficheros Binarios

Los ficheros binarios almacenan secuencias de dígitos binarios que nos son legibles directamente por el usuario como ocurría con los ficheros de texto. Tienen la ventaja de que ocupan menos espacio en disco. En Java, las dos clases que nos permiten trabajar con ficheros son `FileInputStream` (para entrada) y `FileOutputStream` (para salida), estas trabajan con flujos de bytes y crean un enlace entre el flujo de bytes y el fichero.

Escritura de Datos Primitivos Java en un Fichero

Para almacenar datos en un fichero en forma de tipos básicos Java, el paquete `java.io` proporciona las clases `FileOutputStream` y `DataOutputStream`.

El procedimiento es similar al empleado para la clase `FileWriter`, el primer paso es crear un objeto `FileOutputStream` que permita añadir información al fichero o sobrescribirla, para ello utilizamos los **constructores**:

```
FileOutputStream(File fichero, boolean append);  
FileOutputStream(String path, boolean append);
```

La clase `FileOutputStream` es una subclase de `OutputStream`, que representa un stream o flujo de salida para la escritura de bytes.

El segundo paso es a partir del objeto `FileOutputStream` crear un objeto `DataOutputStream` para realizar la escritura. El **constructor** es:

```
DataOutputStream(OutputStream);
```

La clase `DataOutputStream` proporciona métodos para escribir datos en un fichero en cada uno de los ocho **tipos primitivos Java**. Estos métodos tienen el formato:

```
void writeXnn(xxx dato);
```

siendo `xxx` el nombre del tipo primitivo Java.

Métodos para Escritura
<code>void writeBoolean(boolean v);</code>
<code>void writeByte(int v);</code>
<code>void writeBytes(String s);</code>
<code>void writeShort(int v);</code>
<code>void writeChars(String s);</code>
<code>void writeChar(int v);</code>
<code>void writeInt(int v);</code>
<code>void writeLong(long v);</code>
<code>void writeFloat(float v);</code>
<code>void writeDouble(double v);</code>
<code>void writeUTF(String str);</code>

El siguiente ejemplo escribe números decimales en un fichero binario, todos los datos son del mismo tipo

Unidad didáctica 1 MANEJO DE FICHEROS.

```
package ejemplos05EscrituraBinarios;

import java.io.*;
//ejemplo que escribe numeros doubles en un fichero
// todos los números son reales

public class Ej01FileOutputStreamDoubles {
    public static void main(String[] args) {
        // declaramos los objetos FileOutputStream y
        DataOutputStream
        FileOutputStream fe;
        DataOutputStream ds = null;
        try{
            // array de enteros
            double [] array = {5.25, 145.18, 2.37, 143.12, 147.10,
169.19, 47.65};

            fe = new FileOutputStream("Decimales.txt", false);
            ds = new DataOutputStream(fe);
            // mientras el array tenga elementos los escribimos en
            el fichero

            for (double i : array) {
                ds.writeDouble(i);
            }
        }catch(FileNotFoundException e){
            System.out.println("No se pudo abrir el fichero
Enteros.txt");
        }catch(IOException e){
            System.out.println("No se pudo escribir en el fichero
Enteros.txt");
        }finally{

            try {
                ds.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }
    }
}
```

Unidad didáctica 1 MANEJO DE FICHEROS.

Ejemplo escritura en un fichero de dos arrays, uno de enteros y el segundo de String.

```
package ejemplos05EscrituraBinarios;

import java.io.*;

public class Ej02FileOutputStreamEnterosString {

    //ejemplo que escribe un numero entero y un String

    public static void main(String[] args) {
        // declaramos los objetos FileOutputStream y
        DataOutputStream
        FileOutputStream fe = null;
        DataOutputStream ds = null;
        try{

            // array de enteros
            int [] array = {5, 18, 23, 12, 10, 1, 47};
            String [] nombres = {"Beatriz", "Ana", "Maria",
"Juan", "Pedro", "Teresa",
                                "Marta"};

            fe = new FileOutputStream("EnterosNombres.txt", true);
            ds = new DataOutputStream(fe);
            // mientras el array tenga elementos los escribimos en
            el fichero
            for (int i=0 ;i<array.length; i++) {
                ds.writeInt(array[i]);
                ds.writeUTF(nombres[i]);
            }
        }catch(IOException e){
            System.out.println("No se pudo escribir en el fichero
Enteros.txt");
        }
        finally{
            try{
                ds.close();
                fe.close();
            }
            catch(IOException e){
                System.out.println("No se pudo cerrar el fichero
Enteros.txt");
            }
        }

        System.out.println("Fin del programa");
    }
}
```

Unidad didáctica 1 MANEJO DE FICHEROS.

Los métodos que proporciona la clase **FileOutputStream** para escritura son:

Método	Función
void write(int b)	Escribe un byte
void write(byte[] b)	Escribe b.length bytes
void write (byte[] b, int desplazamiento, int n)	Escribe n bytes a partir de la matriz de bytes de entrada y comenzando por b [desplazamiento]

En el siguiente ejemplo se escriben números enteros.

```
package ejemplos05EscrituraBinarios;
```

```
import java.io.*;
```

```
public class Ej03FileOutputStreamWrite {
    public static void main(final String[] args) {
        FileOutputStream fos = null;
        try{
            // array de enteros
            int [] array = {55, 185, 237, 142, 150, 21, 487};

            fos = new FileOutputStream("Enteros01.txt", false);
            // mientras el array tenga elementos los escribimos en
            el fichero
            for (int i : array) {
                fos.write(i);
            }
        }
        catch(IOException e){
            System.out.println("No se pudo escribir en el fichero
            Enteros.txt");
        }
        finally{
            try{
                fos.close();
            }
            catch(IOException e){
                System.out.println("No se pudo cerrar el fichero
                Enteros.txt");
            }
        }
        System.out.println("Fin del Programa");
    }
}
```

Unidad didáctica 1 MANEJO DE FICHEROS.

Lectura de Tipos Primitivos en un Fichero

Para leer datos de un fichero en forma de tipos básicos Java, el paquete `java.io` proporciona las clases `FileInputStream` y `DataInputStream`.

El procedimiento es similar al empleado para la clase `FileReader`, el primer paso es crear un objeto `FileInputStream` que permita recuperar información del fichero, para ello utilizamos los **constructores**:

```
FileInputStream(File fichero);
```

```
FileInputStream (String path);
```

La clase `FileInputStream` es una subclase de `InputStream`, que representa u stream o flujo de entrada para la lectura de bytes.

El segundo paso es a partir del objeto `FileInputStream` crear un objeto `DataInputStream` para realizar la escritura. El **constructor** es:

```
DataInputStream(Input Stream);
```

La clase `DataInputStream` proporciona métodos para leer datos desde un fichero en cada uno de los ocho tipos primitivos Java. Estos métodos tienen el formato:

```
void readXnn(xxx dato);
```

siendo `xxx` el nombre del tipo primitivo Java.

Métodos para Lectura
<code>boolean readBoolean();</code>
<code>byte readByte();</code>
<code>int readUnsignedByte();</code>
<code>int readUnsignedShort();</code>
<code>short readShort();</code>
<code>char readChar();</code>
<code>int readInt();</code>
<code>long readLong();</code>
<code>float readFloat();</code>
<code>double readDouble();</code>
<code>String readUTF();</code>

Hay una excepción **EOFException** cuando intentamos leer de un fichero que ya no tiene información, que es la que captura que hemos llegado al final del fichero.

El siguiente ejemplo lee los datos del fichero `Decimales.txt` que contiene números reales.

```
package ejemplos06LecturaBinarios;
/*
 * Lee numeros reales del fichero Decimales.txt
 */

import java.io.*;

public class Ej01FileInputStreamDoubles {
```

Unidad didáctica 1 MANEJO DE FICHEROS.

```
public static void main(String[] args) {

    FileInputStream fis = null;
    DataInputStream das = null;
    double numero;

    try {
        fis = new FileInputStream("Decimales.txt");
        das = new DataInputStream(fis);

        //leemos los datos mientras existan
        while (true){
            System.out.println ("Numero: "+
das.readDouble());
        }
    }catch (FileNotFoundException fnf){
        System.out.println("No se ha podido encontrar el
fichero " );
    }catch (EOFException eof){
    }catch (IOException ioe){
        System.out.println("No se ha podido leer la
información del fichero " );
    }finally{
        try{
            das.close();
            fis.close();
        }catch (IOException ioe){
            System.out.println("No se ha podido cerrar el
flujo del fichero " );
        }
    } //fin finally
}
}
```

El siguiente ejemplo lee el fichero binario EnterosNombres.txt que contiene números enteros y String.

```
package ejemplos06LecturaBinarios;
import java.io.*;

public class Ej02FileInputStreamIntString {
    public static void main(final String[] args) {
        FileInputStream fLectura = null;
        DataInputStream ds = null;
        try{
            fLectura = new FileInputStream("EnterosNombres.txt");
            ds = new DataInputStream(fLectura);
            //bucle infinito
            while(true){
                System.out.print(ds.readInt() + " " );
            }
        }
    }
}
```

Unidad didáctica 1 MANEJO DE FICHEROS.

```
        System.out.println(ds.readUTF());
    }
} catch (FileNotFoundException e){
    System.out.println("ERROR GRAVE: El fichero
Enteros.txt no está disponible");
    return;
} catch (EOFException e){
    System.out.println("Fin del fichero");
} catch (IOException ioe){
    System.out.print("Error no se ha podido leer de
Enteros.txt");
}
finally{
    try{
        ds.close();
    }
    catch (IOException e){
        System.out.print("Error no se ha podido cerrar
Enteros.txt");
    }
}
}
System.out.println("El programa continua si hubiese mas
sentencias");
}
```

Ejemplo de Lectura/Escritura en un fichero Binario.

```
package ejemplos06LecturaBinarios;
```

```
/*
 * Ejemplo que escribe y lee datos binarios en un fichero
 * 1º Nos pide el nombre del fichero donde va a escribir o leer
 * 2º Nos pide si queremos abrir el fichero en el modo sobrescribir o
insertar
 * 3º Introducimos los datos en el fichero (codigo y nombre de
usuario)
 */
```

```
import java.io.*;
```

```
import aaIntroducirDatos.IntroducirDatos;
```

```
public class Ej03FileInputStream {
```

```
    static String nbFichero;
```

```
    public static void main(final String[] args) {
```

```
        nbFichero = nombreFichero();
```

Unidad didáctica 1 MANEJO DE FICHEROS.

```
int opcion = 0;
do{
    try{
        System.out.println("1.- Escribir en el fichero");
        System.out.println("2.- Leer del fichero");
        System.out.println("3.- Salir");

        opcion =
Integer.parseInt(IntroducirDatos.introducirDatos("Elegir opción: "));

        switch (opcion){
            case 1:
                escrituraDisco();
                break;
            case 2:
                lecturaDisco();
                break;
            case 3:
                System.exit(0);
            default:
                System.out.println("Error en la opcion");
        }
    }catch(NumberFormatException e){
        System.out.println("Error al introducir la
opcion");
    }

    }while(opcion != 3);
} // fin del main

private static String nombreFichero() {
    nbFichero = IntroducirDatos.introducirDatos("Nombre del
fichero: ");
    return nbFichero;
}

// metodo para escribir en el disco los datos a introducir
static void escrituraDisco(){
    File f = null;
    DataOutputStream dos = null;

    String respuesta="";

    try{

        //creamos el objeto File
        f = new File (nbFichero) ;
```


Unidad didáctica 1 MANEJO DE FICHEROS.

```
preguntamos // Comprobamos si el fichero existe en cuyo caso
// si se quiere sobrescribir o no
if (f.exists()){
    respuesta= IntroducirDatos.introducirDatos("El
fichero ya existe ¿Desea sobrescribirlo (SI/NO)?");
}
// según sea la respuesta abrimos el fichero en modo
sobrescritura o en el modo añadir

//compareToIgnoreCase es que compare sin distinguir
mayúsculas y minúsculas
if (respuesta.compareToIgnoreCase("si") == 0){
    dos = new DataOutputStream ( new
FileOutputStream(nbFichero));
}else{
    dos= new DataOutputStream (new
FileOutputStream(nbFichero, true));
}

// introducimos los datos que queremos escribir en el
fichero
String nombre=
IntroducirDatos.introducirDatos("Introduce el nombre del usuario por
teclado. "
    + "F Para finalizar: ");
while(!nombre.equalsIgnoreCase("F")){
    // el código lo convertimos a entero
    int codigo =
Integer.parseInt(IntroducirDatos.introducirDatos("Introduce el código
"
        + "del usuario por teclado: "));

    //escribimos los datos en el fichero
    dos.writeInt(codigo);
    dos.writeUTF(nombre);
    nombre=
IntroducirDatos.introducirDatos("Introduce el nombre del usuario por
teclado. F Para finalizar: ");
}
}catch (IOException ioe){
    System.out.println("No se ha podido escribir la
información en el fichero " +f .getName());
}finally{
    try{
        if (dos != null){
            dos.close();
        }
    }catch (IOException ioe){
```

Unidad didáctica 1 MANEJO DE FICHEROS.

```
        System.out.println("No se ha podido cerrar
correctamente el flujo del fichero " + f.getName());
    }
}
} // fin del método escrituraDisco

static void lecturaDisco(){
    File f = null;
    DataInputStream dis =null;

    System.out.println() ;
    try{

        // abrimos el fichero para lectura
        f= new File (nbFichero);
        dis= new DataInputStream (new FileInputStream(f));
        //leemos los datos mientras existan
        //Ojo el orden al leer tiene que ser el mismo que al
escribir

        while (true){
            System.out.println ("Codigo de Usuario: "+
dis.readInt());
            System.out.println ("Nombre de Usuario: "+
dis.readUTF());
        }
    }catch (FileNotFoundException fnf){
        System.out.println("No se ha podido encontrar el
fichero " +f.getName());
    }catch (EOFException eof){
    }catch (IOException ioe){
        System.out.println("No se ha podido leer la
información del fichero " +f.getName());
    }finally{
        try{
            if (dis!=null){
                dis.close();
            }
        }catch (IOException ioe){
            System.out.println("No se ha podido cerrar el
flujo del fichero " + f.getName());
        }
    } //fin finally
} //fin lecturaDisco
} // fin clase
```