

10.C. Ficheros.

1. Trabajando con ficheros.

1.5. Acceso secuencial.

En el siguiente ejemplo vemos cómo se **escriben datos en un fichero secuencial**: el nombre y apellidos de una persona utilizando el método `writeUTF()` que proporciona `DataOutputStream`, seguido de su edad que la escribimos con el método `writeInt()` de la misma clase. A continuación escribimos lo mismo para una segunda persona y de nuevo para una tercera. Después cerramos el fichero. Y ahora lo abrimos de nuevo para ir **leyendo de manera secuencial** los datos almacenados en el fichero, y escribiéndolos a consola.

```
/*  
  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package escylecsecuen;  
  
import java.io.DataInputStream;  
import java.io.DataOutputStream;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
/**  
 *  
 * @author JJBH  
 */  
  
public class escylee {  
  
    /**  
     * @param args the command line arguments  
     */  
  
    public static void main(String[] args) {  
  
        // Declarar un objeto de tipo archivo  
  
        DataOutputStream archivo = null ;  
        DataInputStream fich = null ;  
        String nombre = null ;  
        int edad = 0 ;  
  
        try {  
  
            // Creando o abriendo para añadir el archivo  
  
            archivo = new DataOutputStream( new FileOutputStream("c:\\secuencial.dat",true) );
```

```
// Escribir el nombre y los apellidos
```

```
archivo.writeUTF( "Antonio López Pérez " );
```

```
archivo.writeInt(33) ;
```

```
archivo.writeUTF( "Pedro Piqueras Peñaranda" );
```

```
archivo.writeInt(45) ;
```

```
archivo.writeUTF( "José Antonio Ruiz Pérez " );
```

```
archivo.writeInt(51) ;
```

```
// Cerrar fichero
```

```
archivo.close();
```

```
// Abrir para leer
```

```
fich = new DataInputStream( new FileInputStream("c:\\secuencial.dat") );
```

```
nombre = fich.readUTF() ;
```

```
System.out.println(nombre) ;
```

```
edad = fich.readInt() ;
```

```
System.out.println(edad) ;
```

```
nombre = fich.readUTF() ;
```

```
System.out.println(nombre) ;
```

```
edad = fich.readInt() ;
```

```
System.out.println(edad) ;
```

```
nombre = fich.readUTF() ;
```

```
System.out.println(nombre) ;
```

```
edad = fich.readInt() ;
```

```
System.out.println(edad) ;
```

```
fich.close();
```

```
} catch(FileNotFoundException fnfe) { /* Archivo no encontrado */ }
```

```
catch (IOException ioe) { /* Error al escribir */ }
```

```
catch (Exception e) { /* Error de otro tipo*/
```

```
System.out.println(e.getMessage());}
```

```
}
```

Fíjate al ver el código, que hemos tenido la precaución de ir escribiendo las cadenas de caracteres con el mismo tamaño, de manera que sepamos luego el tamaño del registro que tenemos que leer.

Por tanto para **buscar información en un fichero secuencial**, tendremos que **abrir el fichero e ir leyendo registros hasta encontrar el registro que buscamos**.

¿Y si queremos **eliminar un registro en un fichero secuencial**, qué hacemos? Esta operación es un problema, puesto que no podemos quitar el registro y reordenar el resto. Una opción, aunque costosa, **sería crear un nuevo fichero. Recorremos el fichero original y vamos**

copiando registros en el nuevo hasta llegar al registro que queremos borrar. Ese no lo copiamos al nuevo, y seguimos copiando hasta el final, el resto de registros al nuevo fichero. De este modo, obtendríamos un nuevo fichero que sería el mismo que teníamos pero sin el registro que queríamos borrar. Por tanto, si se prevé que se va a borrar en el fichero, no es recomendable usar un fichero de este tipo, o sea, secuencial.

Autoevaluación

Señala si es verdadera o es falsa la siguiente afirmación:

Para encontrar una información almacenada en la mitad de un fichero secuencial, podemos acceder directamente a esa posición sin pasar por los datos anteriores a esa información. ¿Verdadero o Falso?

- ☐ Verdadero.
- ☒ Falso.