

10.C. Ficheros.

1. Trabajando con ficheros.

1.2. Ficheros binarios y ficheros de texto (I).

Ya comentamos anteriormente que los ficheros se utilizan para guardar la información en un soporte: disco duro, disquetes, memorias usb, dvd, etc., y posteriormente poder recuperarla. También distinguimos dos tipos de ficheros: los de **texto** y los **binarios**.

En los **ficheros de texto** la información se guarda como caracteres. Esos caracteres están codificados en **Unicode**, o en **ASCII** u otras codificaciones de texto.

En la siguiente porción de código puedes ver cómo para un fichero existente, que en este caso es texto.txt, averiguamos la codificación que posee, usando el método `getEncoding()`

```
FileInputStream fichero;
try {
    // Elegimos fichero para leer flujos de bytes "crudos"
    fichero = new FileInputStream("c:\\texto.txt");
    // InputStreamReader sirve de puente de flujos de byte a caracteres
    InputStreamReader unReader = new InputStreamReader(fichero);
    // Vemos la codificación actual
    System.out.println(unReader.getEncoding());
} catch (FileNotFoundException ex) {
}
}
```

El mismo código copiable:

```
FileInputStream fichero;

try {

    // Elegimos fichero para leer flujos de bytes "crudos"

    fichero = new FileInputStream("c:\\texto.txt");

    // InputStreamReader sirve de puente de flujos de byte a caracteres

    InputStreamReader unReader = new InputStreamReader(fichero);

    // Vemos la codificación actual

    System.out.println(unReader.getEncoding());

} catch (FileNotFoundException ex) {

    Logger.getLogger(textos.class.getName()).log(Level.SEVERE, null, ex);

}
```

Para **archivos de texto**, se puede abrir el fichero para leer usando la clase `FileReader`. Esta clase nos proporciona métodos para **leer caracteres**. Cuando nos interese no leer carácter a carácter, sino **leer líneas completas**, podemos usar la clase `BufferedReader` a partir de `FileReader`. Lo podemos hacer de la siguiente forma:

```
File arch = new File ("C:\\fich.txt");

FileReader fr = new FileReader (arch);

BufferedReader br = new BufferedReader(fr);

...

String linea = br.readLine();
```

Para **escribir en archivos de texto** lo podríamos hacer, teniendo en cuenta:

```
FileWriter fich = null;
```

```
PrintWriter pw = null;
```

```
fich = new FileWriter("c:/fich2.txt");
```

```
pw = new PrintWriter(fichero);
```

```
pw.println("Linea de texto");
```

```
...
```

Si el fichero al que queremos escribir existe y lo que queremos es añadir información, entonces pasaremos el segundo parámetro como true:

```
FileWriter("c:/fich2.txt",true);
```

Para saber más

En el siguiente enlace a wikipedia puedes ver el código ASCII.

[Código Ascii.](#)