

UA01

5.3. Flujo basado en caracteres

Para los flujos de caracteres, Java dispone de dos clases abstractas: **Reader** y **Writer**.

Si se usan sólo **FileInputStream**, **FileOutputStream**, **FileReader** o **FileWriter**, cada vez que se efectúa una lectura o escritura, se hace físicamente en el disco duro. Si se leen o escriben pocos caracteres cada vez, el proceso se hace costoso y lento por los muchos accesos a disco duro.

Los **BufferedReader**, **BufferedInputStream**, **BufferedWriter** y **BufferedOutputStream** añaden un **buffer** intermedio. Cuando se lee o escribe, esta clase controla los accesos a disco. Así, si vamos escribiendo, se guardarán los datos hasta que haya bastantes datos como para hacer una escritura eficiente.

Al leer, la clase leerá más datos de los que se hayan pedido. En las siguientes lecturas nos dará lo que tiene almacenado, hasta que necesite leer otra vez físicamente. Esta forma de trabajar hace los accesos a disco más eficientes y el programa se ejecuta más rápido.

En un programa Java para crear o abrir un fichero se invoca a la clase **File** y a continuación, se crea el flujo de entrada hacia el fichero con la clase **FileReader**. Después se realizan las operaciones de lectura o escritura y cuando terminemos de usarlo lo cerraremos mediante el método **close()**.

Ejemplo 1

El siguiente ejemplo lee cada uno de los caracteres del fichero de texto de nombre **LeerFichTexto.java** y los muestra en pantalla, los métodos **read()** pueden

lanzar la excepción **IOException**, por ello en **main()** se ha añadido **throws IOException** ya que no se incluye el manejador **try-catch**.

Observa que la expresión **((char) i)** convierte el valor entero recuperado por el método **read()** a carácter, es decir, hacemos un cast a **char**. Se llega al final del fichero cuando el método **read()** devuelve **-1**.

Ejemplo 2

El siguiente ejemplo escribe caracteres en un fichero de nombre **FichTexto.txt** (si no existe lo crea). Los caracteres se escriben uno a uno y se obtienen de un **String**:

EscribirFichTexto.java

En vez de escribir los caracteres uno a uno, también podemos escribir todo el array: **fic.write(cad);**

Obra publicada con **Licencia Creative Commons Reconocimiento Compartir igual 4.0**
<<http://creativecommons.org/licenses/by-sa/4.0/>>