

Introducción

En esta parte del tema se van a utilizar varios conceptos que ya os sonarán repetidos si habéis cursado el módulo de programación.

Como comentamos anteriormente, cuando apagamos el ordenador, los datos de la memoria RAM se pierden. Para evitar esto, será necesario almacenar esa información en un fichero de tal forma que esta **persista** una vez el ordenador sea apagado.

En esta unidad veremos, entre otras cosas, cómo hacer con Java las operaciones de crear, actualizar y procesar ficheros.

OPERACIONES DE ENTRADA Y SALIDA

Denominamos operaciones de entrada y salida a aquellas operaciones que constituyen un flujo de información del programa con el exterior.

Las operaciones de E/S en Java las proporciona el paquete estándar de la API de Java denominado **java.io** que incorpora interfaces, clases y excepciones para acceder a todo tipo de ficheros.

Librería *java.io*

Esta librería contiene las clases necesarias para gestionar las operaciones de entrada y salida con Java.

Estas clases de E/S las podemos agrupar fundamentalmente en:

- Clases para **leer entradas desde** un flujo de datos.
- Clases para **escribir entradas a** un flujo de datos.
- Clases para **operar con ficheros en el sistema** de ficheros local.
- Clases para **gestionar la serialización** de objetos.

Por otro lado, otra librería también conocida es **java.nio (Non- Blocking I/O)**.

Esta librería fue introducido en el API de Java desde la versión 1.4 como extensión eficiente a los paquetes java.io y java.net.

Java NIO ofrece una forma de trabajar con IO (Input / Output) diferente a la utilizada en la API de IO estándar ya que esta se basa en los conceptos de **Buffer** y el **Channel**.

Diferencias entre java.io y java.nio

La utilización de una librería u otra dependerá de lo que necesitemos. Podemos elegir entre una u otra o bien combinarlos de forma conjunta.

Las principales diferencias entre ellas son:

java.io	java.nio
Orientado a Stream	Orientado a buffered
Procesa datos en stream	Procesa datos en bloques
Operaciones E/S bloqueantes	Operaciones E/S no bloqueantes
Canales no disponibles	Canales disponibles
No dispone del concepto de selector	Dispone del concepto de selector

A continuación, hablaremos en más detalle en cada una de estas características:

1. Orientados a Stream vs Orientado a buffer.

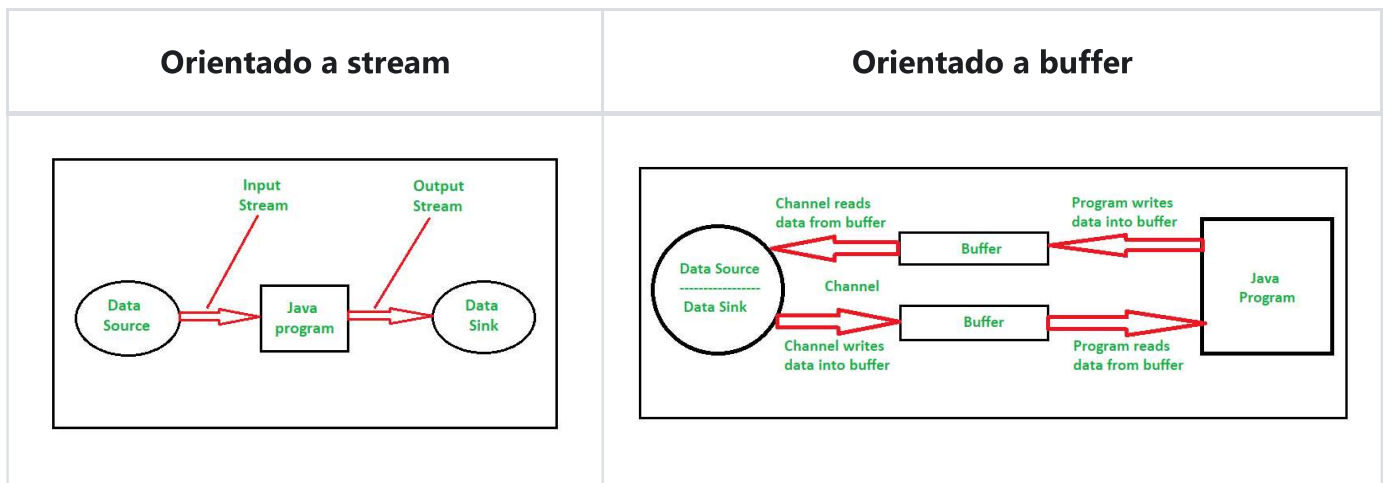
o **Orientado a stream:**

- Solo se puede leer uno o más bytes al mismo tiempo del stream.
- Se utilizan stream para transferir datos entre la fuente/destino y el programa java.
- La transferencia en una única dirección.

o **Orientado a buffer:**

- Se pueden leer bloques de bytes.
- La información se lee de un buffer desde la cual se procesa utilizando un canal.
- La transferencia es bidireccional.

- o En las siguientes imágenes se puede comparar visualizar las diferencias:



2. Operaciones E/S bloqueantes o no bloqueantes:

- Una **operación de E/S es bloqueante** cuando el hilo que la ejecuta es bloqueado hasta que la lectura/escritura es finalizada, es decir, es una **transferencia síncrona**.
- Una **operación de E/S es no bloqueante** cuando el hilo que ejecuta la operación puede hacer otra tarea intermedia mientras espera a que la operación finaliza, es decir, es una **transferencia asíncrona**.

3. **Canales:** un canal es un medio de transmisión de información eficiente entre una entidad (programa) y el buffer.

4. **Selector:** permite seleccionar un canal de entre todos los posibles canales de E/S usando un único.

java.io.File y java.nio.File

Dentro de las librerías anteriores, la clase File es la utilizada para realizar las operaciones E/S con ficheros.

Al principio se utilizaba la clase **java.io.File**, sin embargo, esta presenta ciertas limitaciones:

- **Complicado detectar y resolver errores:** Muchos métodos no lanzaban excepciones.
- **No permite manejar enlaces simbólicos**
- Presenta **problemas de escalabilidad:** los directorios largos provocan problemas de memoria e incluso de denegación de servicio.

Para solventar estas limitaciones surge la clase **java.nio.File**.

! PARA AMPLIAR

En un sistema de archivos podemos crear **enlaces** que nos permitan acceder a archivos o carpetas de forma sencilla, sin la necesidad de desplazarnos por toda la jerarquía de directorios.

Podemos diferenciar dos tipos de enlaces:

1. Enlace físico (Duro):

- Es un nombre más para un fichero en otra ubicación.
- El fichero solo es eliminado del disco duro cuando se haya eliminado el archivo y todos los enlaces físicos a él.
- Será el mismo fichero en todas las ubicaciones donde haya un enlace físico.
- Solo pueden ser creados para fichero y no para directorios.

2. Enlace simbólico (Blando):

- Es un puntero a un fichero o directorio.
- Si el archivo original es eliminado el enlace simbólico queda inutilizable.

Si queréis saber más sobre la creación de estos enlaces [aquí](#) tenéis más información.