

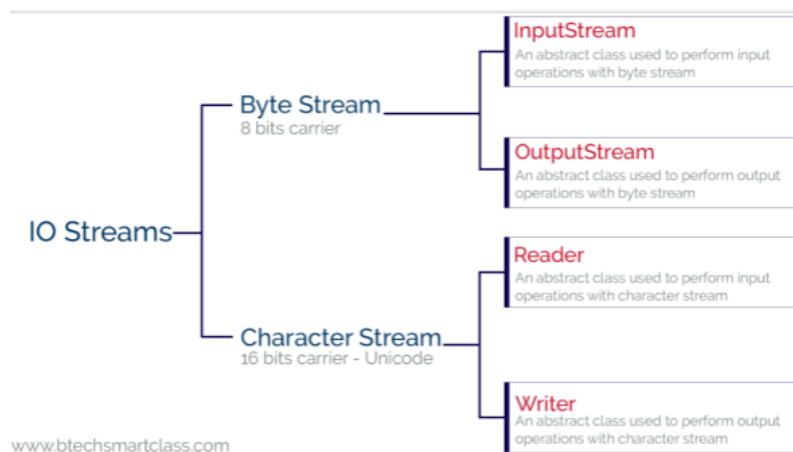
EJERCICIOS UNIDAD 1

3. RA1 -CE f) Comunicación de datos entre procesos

En Java, los procesos tienen un sistema de *streams* (flujos) para compartir información:

- Entrada estándar (**stdin**):
 - Stream donde el proceso lee datos de entrada.
 - Origen: otro proceso, fichero, NIC...
- Salida estándar (**stdout**):
 - Stream donde el proceso escribe (comparte) datos.
 - Destinos: impresora, fichero, pantalla, otro proceso...
- Salida de error estándar (**stderr**):
 - Stream donde el proceso escribe mensajes de error.
 - Destino: Sytem.err

Estas son las clases definidas en el paquete **java.io**



Ejercicios comunicación de datos entre procesos

9. En este ejercicio el estudiante debe realizar un ejercicio que lance el comando:

- Imprime el sistema operativo donde estamos (ver en enlace del ejercicio 2).
- Esperamos 5 segundos.
- Si estamos en Windows, abre ipconfig e imprime su salida por pantalla y en un fichero.
- Si estamos en Linux o Mac OS abrimos ifconfig e imprimimos su salida por pantalla y en un fichero.
- Esperamos tres segundos.
- Imprimimos “Las redes parecen que van bien”.

ATENCIÓN!!: Modifica haz otra versión del ejercicio utilizando el bloque “try-with-resources”.

ATENCIÓN2!!: Añade al ejercicio anterior el código necesario para imprimir también la salida de error estándar **stderr**.

CE	No realizado o muy mal	Insuficiente	Suficiente	Notable	Sobresaliente
CE e) Se han utilizado clases para programar aplicaciones que crean subprocesos.	No se crea ningún subproceso.	Se intenta crear un subproceso, pero erróneamente.	Se crea el subproceso sólo de un sistema operativo.	Se crean los subprocesos tanto en Linux como en Mac, pero hay algún error pequeño en la creación de alguno de ellos.	Se crean los subprocesos correctamente tanto en Windows como en Linux o Mac.
CE f) Se han utilizado mecanismos para sincronizar y obtener el valor devuelto por los subprocesos iniciados.	No se obtiene la salida de los subprocesos.	Se intenta obtener la salida de los subprocesos, pero erróneamente.	Se obtiene la salida de los dos subprocesos y no se hace ninguna modificación propuesta.	Se obtiene la salida de los dos subprocesos y se hace una de las modificaciones propuestas.	Se obtiene la salida de los subprocesos y se hacen las dos modificaciones propuestas.

10. Obtener la salida estándar de un .jar.

Crea dos programas en Java:

- El **primero** imprime por su salida estándar 10 números aleatorios. A continuación, genera un .jar del programa.
- El **segundo** tiene que ejecutar el .jar del primer programa y leer su salida estándar para calcular la suma de los 10 números aleatorios.

11. Escribir en la entrada estándar de un proceso.

Crea dos programas en Java:

- El **primero** lee dos cadenas de texto desde la entrada estándar. Para hacerlo utiliza la clase `BufferedReader` y su método `readLine()`. Para redirigir la entrada estándar (`System.in`) a `BufferedReader` hay que hacer el código siguiente:

```
| | InputStreamReader insr = new InputStreamReader (System.in); //System.in = Stream
| | standar de entrada, habitualmente el teclado
| | BufferedReader br = new BufferedReader (insr);
```

Pídelas al usuario con los mensajes: “Introduce la primera cadena” e “introduce la segunda cadena”.

Cada cadena leída la tienes que imprimir por pantalla (salida estándar).

Por último, crea un .jar de la aplicación.

- El **segundo** programa debe lanzar un proceso con el .jar anterior y debe pasarle por su entrada estándar las dos cadenas para que las lea. A continuación, comprobar que funciona deberás obtener la salida estándar del proceso para imprimirla por pantalla.