

EJERCICIOS - UNIDAD 1

1. Muestra el entorno de ejecución de un proceso con `Map<String, String> environment()` de `ProcessBuilder`. Hay que **iterar** sobre el **`Map<String, String>`** devuelto para mostrar cada entrada.
2. Crea un programa que **lance** la clase **`ProcesoLento.java`** (para ello, mete en el mismo paquete la clase `ProcesoLento.java` y la clase que vayas a crear). Utiliza el método **`isAlive()`** de la clase **`Process`** para **comprobar** si la clase `ProcesoLento` se sigue **ejecutando** e imprimir un mensaje indicando que el proceso sigue en ejecución. Esta comprobación debe hacerla **cada 3 segundos** mientras esté en ejecución. Cuando ya no esté en ejecución debe terminar, para lo cual utiliza **`Thread.sleep(int tiempo_ms)`**.
3. Crea una clase Java que lance un proceso haciendo uso de la clase `ProcessBuilder`. En vez de tomar el comando por los argumentos del main, debe **solicitarle al usuario por consola** que introduzca el **comando/programa a ejecutar**.

REDIRECCIÓN ENTRADA/SALIDA CON PROCESSBUILDER

4. Partiendo del ejercicio 2, modifícalo para redireccionar la salida estándar del proceso a un fichero de texto llamado **`salidaProcesoLento.txt`**.
5. Crea una clase Java llamada **`PreguntaNombre.java`**. Esta clase debe preguntarle al usuario su nombre, el cual debe introducirlo por la entrada estándar, e imprimir en la salida estándar “¡Hola *nombre*!”, donde nombre es el nombre introducido por teclado. Crea **otra clase** que se encargue de **lanzar `PreguntaNombre.java` como un proceso**. Además, **redirecciona la entrada estándar** para que se lea desde un **fichero**. La **salida** del proceso debe ser la estándar, es decir, la **pantalla**.
6. Crea una clase Java llamada **`TicketCompra.java`**. Esta clase debe pedirle al usuario los siguientes datos:
 - **Código del producto:** debe ser un número entero que represente el código del producto.
 - **Precio del producto:** debe ser un número decimal que represente el precio del producto.
 - **Cantidad:** debe ser un número entero que indique el número de unidades que se han comprado de ese producto.

(No imprimir mensaje en la salida estándar para que pida los distintos datos).

En la salida estándar debe imprimirse los datos introducidos por el usuario de la siguiente manera:

Código - Precio - Cantidad - Total

Donde *Total* es un valor calculado a partir del *Precio* y la *Cantidad*.

Crea **otra clase** que se encargue de **lanzar TicketCompra.java** como un **proceso**. Además, **redirecciona** la **salida estándar** para que escriba la salida estándar **en un fichero**. Cada vez que escriba en el fichero debe **concatenar** el nuevo contenido, **no sobreescribir** el anterior. La **entrada** del proceso debe ser la estándar, es decir, el **teclado**.

7. En este ejercicio vamos a lanzar varios procesos, cuyas entradas y salidas están enlazadas mediante tuberías. Para lo que usaremos el método **startPipeline**. Para ello, vamos a crear 3 clases distintas que serán los distintos procesos a lanzar:
 - **Clase 1:** Va a generar 10 direcciones IP de forma aleatoria y las va a imprimir por la salida estándar.
 - **Clase 2:** Va a leer 10 direcciones IP de teclado y va a imprimir por consola únicamente aquellas que pertenezcan a las clases A, B o C.
 - **Clase 3:** Va a leer una serie de direcciones IP por teclado (no sabemos cuántas van a llegar) y va a imprimir por consola la dirección IP y a continuación la clase a la que pertenece.

Tienes que crear una clase que **lance estos tres procesos** en el orden en el que se especifican, de forma que la salida estándar de uno sea la entrada estándar del siguiente. La **salida estándar** del **último proceso** debe ser un **fichero** de texto.

CONSEJO: Establece para los tres procesos la **salida de error estándar**, para en caso de que haya algún error durante la ejecución, éste se pinte en la consola.

REDIRECCIÓN A STREAMS DE LA CLASE PROCESS

8. Al igual que se hizo en el ejercicio 4, crea una clase que lance el proceso **ProcesoLento.Java** pero usando la clase **Runtime** en vez de la clase **ProcessBuilder**. En este ejercicio vamos a tomar la salida del proceso con el método **getInputStream()** de la clase **Process**, y la vamos a leer para luego escribirla en un fichero llamado **salidaProcesoLento2.txt**.
9. Realiza de nuevo el ejercicio 5, pero en vez de utilizar **ProcessBuilder** para lanzar la clase **PreguntaNombre.java** utiliza la clase **Runtime**. Utiliza el método **getOutputStream()** de la clase **Process** para que tome los datos de un fichero de texto, por ejemplo, **entradaNombre.txt**. Para poder ver la salida en la consola habrá que usar el método **getInputStream()** e imprimirlo por pantalla.

SINCRONIZACIÓN ENTRE PROCESOS

10. Crea una clase que lance los siguientes procesos y que sincronice la ejecución entre ellos:
- **Proceso 1:** Creación por línea de comandos de una carpeta de nombre “**carpeta1**” en la ruta C:\pruebas. Si esta carpeta no existe créala.
 - **Proceso 2:** Creación por línea de comandos un fichero en blanco en la carpeta creada en el Proceso 1. Llama al fichero “**fichero1.txt**”. Deberá esperar a que termine correctamente el Proceso 1 antes de crear el fichero.
 - **Proceso 3:** Abre el fichero creado en el Proceso 2 con el bloc de notas (el comando es **notepad**). Antes de abrir el fichero tiene que esperar a que termine el Proceso 2.
11. Crea una clase que lance los siguientes procesos y que sincronice la ejecución entre ellos:
- **Proceso 1:** Creación por línea de comandos un fichero en blanco en la carpeta “**carpeta1**” que se creó en el ejercicio anterior. Llama al fichero “**fichero2.txt**”.
 - **Proceso 2:** Abre el fichero creado en el Proceso 1 con el bloc de notas (el comando es **notepad**). Antes de abrir el fichero tiene que esperar a que termine el Proceso 1.
 - **Proceso 3:** Debe **lanzar una clase Java** que se encargue de **leer** el contenido del fichero “**fichero2.txt**” y mostrar el contenido del fichero por pantalla. Antes de lanzar esta clase Java debe esperar a que termine de ejecutarse el Proceso 2.

Cuando pongas en ejecución esta clase, en el momento en el que se abra el bloc de notas debes escribir un mensaje, que será el que se muestre cuando se lance el Proceso 3. Hasta que no cierres el bloc de notas no va a ejecutarse el Proceso 3.