TEMAS 1-2

PRÁCTICA 1

1. Hemos visto en los ejemplos que podemos lanzar un proceso con ProcessBuilder y ejecutar comandos de consola (cmd) en él. En el ejemplo indicábamos exactamente el comando a ejecutar y mostrábamos por consola el resultado de la ejecución. La clase ProcessBuilder permite redirigir la entrada, salida y los errores en la ejecución de proceso a ficheros. Busca información en la ayuda de Java sobre los métodos redirectInput, redirectOutput y redirectError.

   Deberás desarrollar un programa que lance un subproceso *cmd* con ProcessBuilder, el programa debe obtener los comandos a ejecutar por la consola de un fichero .bat que habrás creado tú previamente. El programa dejará el log de ejecución en un fichero de salida y el log de errores en otro fichero. Deberás utilizar los métodos redirectInput, redirectOutput y redirectError.

   El fichero bat podría tener por ejemplo los siguientes comandos:

   ping www.dam2chomon.org

   ping www.google.es

   pring www.iesch.org

   De esa forma veremos que ocurre en cada situación:

   - un comando correcto con una dirección que no existe
   - un comando correcto con una dirección que existe
   - un comando incorrecto

2. Crea una aplicación que conste de 2 hilos; el primero el hilo principal de la aplicación Java. El hilo principal deberá lanzar un nuevo hilo encargado de imprimir por consola los siguientes mensajes con un intervalo de 4 segundos entre cada uno de ellos (Mensajes: “Programas”, “Procesos”, “Servicios”, “Hilos”). El hilo principal debe quedar a la espera hasta que termine, mostrando cada segundo que está esperando por la finalización del hilo hijo. La ejecución del programa debe durar 16s ya que son 4 mensajes y 4s de espera por cada uno.

   Para poder reducir la duración de la ejecución el programa debe aceptar por parámetro (método *main*) el tiempo de espera máximo que el hilo principal esperará a la ejecución del hilo secundario, una vez superado ese tiempo el hilo principal debe interrumpir la ejecución del hilo secundario y a partir de ese momento el hilo secundario mostrará los mensajes restantes sin esperas entre la impresión de los mensajes para finalizar la ejecución del hilo cuanto antes. El hilo principal debe mostrar por consola el mensaje de finalización de la ejecución del programa. Puedes imprimir los mensajes que consideres oportunos para verificar la correcta ejecución del programa. Calcula el tiempo de ejecución del hilo

principal y muéstralo por consola. Incluye el nombre del hilo que imprime por consola cada vez que muestres un mensaje de salida.

Ejemplos salida por consola:

```
--Con un tiempo de espera de 3s

Hilo: main. Tiempo de espera: 3s
Hilo: main. Esperando a que el hilo Thread-0 termine
Hilo: main. Todavía esperando...
Hilo: Thread-0. Programas
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Cansado de esperar
Hilo: Thread-0. Procesos
Hilo: Thread-0. Servicios
Hilo: Thread-0. Hilos
Hilo: Thread-0. *** Finalizado ***
Hilo: main. *** Finalizado. Tiempo de ejecución:3s. ***
```

```
--Con un tiempo de espera mayor a 16s, por ejemplo 100

Hilo: main. Tiempo de espera: 100s
Hilo: main. Esperando a que el hilo Thread-0 termine
Hilo: main. Todavía esperando...
Hilo: Thread-0. Programas
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: Thread-0. Procesos
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: Thread-0. Servicios
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: Thread-0. Hilos
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: main. Todavía esperando...
Hilo: Thread-0. *** Finalizado ***
Hilo: main. *** Finalizado. Tiempo de ejecución:16s. ***
```