EJERCICIOS PROCESOS JAVA

Realizar los ejercicios con ProcessBuilder y Runtime.

Actividad 1

Crear una función que se llame *LanzarPrograma* que recibe como parámetro el comando/nombre del programa a ejecutar. Dentro de esta función se debe ejecutar el proceso.

En el programa principal (main) ejecutar un proceso para arrancar la calculadora (*calc*).

Comprobar con PowerShell el ID de proceso de cada uno.

Añadir en un Word capturas del programa creado y funcionamiento, de la búsqueda del ID de proceso con PowerShell (comando utilizado) y de los Ids de proceso encontrados.

Actividad 2

Con el programa anterior llamar a la función LanzarPrograma dos veces para arrancar la aplicación Paint (mspaint)

¿Se crean dos procesos o uno sólo?

Añadir en un Word capturas del programa creado y funcionamiento, de la búsqueda del ID de proceso con PowerShell (comando utilizado) y de los Ids de proceso encontrados.

Actividad 3

Vamos a ejecutar un comando básico como ping y capturar la salida:

- ProcessBuilder: Crea un proceso usando el comando proporcionado (*ping -c 3 google.com*).

  Si os sale el error "Acceso denegado. La opción -c requiere privilegios administrativos", significa que no tienes los permisos necesarios para ejecutar el comando ping con la opción -c en tu sistema. **Utilizar el comando -n (*ping -n 3 google.com)***

- InputStream: Capturamos la salida estándar del proceso

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 4

Crear un archivo ejecutable (*.jar) con el programa de la actividad 3. Crear un nuevo nuevo programa que ejecute ese archivo java: comando (“cmd”, “/c”, “java -jar”, “nombre archivo jar”)

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 5

Modificar el programa de la actividad 4 para que capture tanto la salida estándar como la de error.

Actividad 6

Con el programa anterior llamar a la función LanzarPrograma para arrancar Notepad y la calculadora. El segundo proceso debe esperar a que termine el primero antes de arrancar.

- waitFor(): Espera a que el proceso finalice antes de continuar con el código.

Añadir en un Word capturas del programa creado y funcionamiento, de la búsqueda del ID de proceso con PowerShell (comando utilizado) y de los Ids de proceso encontrados.

Actividad 7

Crear documento de texto en el Escritorio con el siguiente contenido:

“*Hace mucho tiempo, en un pueblo de Italia, vivía un viejo carpintero llamado Gepetto, quien se dedicaba a fabricar juguetes para todos los niños del pueblo, ya que no tenía familia ni nietos propios.*

Gepetto era generoso y compartía con los niños la alegría, pero a veces se sentía solitario y desdichado, por lo que un día reunió un sobrante de madera y decidió crear una simpática marioneta para entretenerse. La obra le salió tan bien y parecía

ser tan real, que el viejo carpintero bautizó al muñeco como Pinocho, mientras suspiraba largamente, deseando que pudiera convertirse en un niño de verdad.”

Utilizar el programa de la actividad 1 para abrir el documento de texto (“cmd", "/c", "start”)

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 8

Crear un programa que muestre por consola las variables de entorno de un proceso.

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 9

Modificar las variables de entorno del proceso antes de ejecutarlo, creando una variable personalizada y mostrarlas por pantalla tras su modificación.

environment.put("MI_VARIABLE", "VALOR_PERSONALIZADO");

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 10

Crear un programa que ejecute el comando dir para listar archivos de un directorio cualquiera y redirija la salida al fichero output.txt.

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 11

Crear un programa que liste los procesos en ejecución desde PowerShell usando el comando Get-Process. Imprimir el resultado por pantalla.

Añadir en un Word capturas del programa creado y funcionamiento.

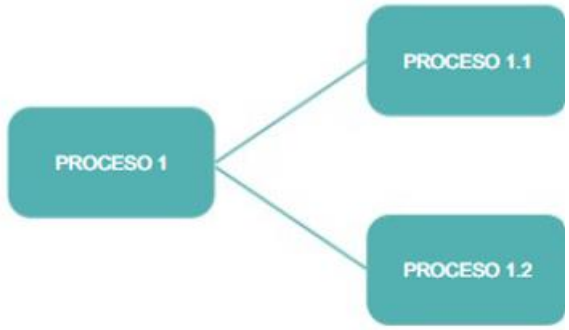
Actividad 12

Crear un programa que inicie un proceso desde PowerShell usando el comando Start-Process. Imprimir el resultado por pantalla.

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 13

Crear tres procesos en Java sincronizados como la figura que se muestra a continuación:



El proceso 1 será el primero en ejecutarse:

- Si el código de finalización del Proceso 1 es 0, se ejecutará el proceso 1.1
- Si el código de finalización del Proceso 1 es distinto de 0, se ejecutará el proceso 1.2

Actividad 14

Crear un programa que finalice un proceso (el que quieras: notepad, navegador) desde PowerShell usando el comando Stop-Process -Name y nos pida confirmación antes de parar el proceso. Imprimir el resultado por pantalla.

Añadir en un Word capturas del programa creado y funcionamiento.

Actividad 15

Crear un programa que muestre el uso de CPU en orden ascendente desde PowerShell usando el comando Get-Process | Sort-Object -Property y redirija la salida al fichero cpu.txt.

Añadir en un Word capturas del programa creado y funcionamiento.