

RELACIÓN DE EJERCICIOS SOBRE PROCESOS EN JAVA

1.- Crea un programa Java llamado *LeerNombre.java* que reciba desde los argumentos de *main()* un nombre y lo visualice en pantalla. Utiliza *System.exit(1)* para una finalización correcta del programa y *System.exit(-1)* para el caso que no se hayan introducido los argumentos correctos en *main()*.

A continuación, haz un programa para ejecutar *LeerNombre.java*. Utiliza el método **waitFor()** para comprobar el valor de salida del proceso que se ejecuta. Prueba la ejecución del programa dando valor a los argumentos de *main()* y sin darle valor. ¿Qué valor devuelve **waitFor()** en un caso y en otro?

2.- Partiendo del ejercicio 1, muestra los errores que se producen al ejecutar un programa que no exista. Muestra los errores en un archivo llamado “errores.txt”.

3.- Supongamos que tenemos un programa Java que lee una cadena desde la entrada estándar y la visualiza:

```
import java.io.*;
public class EjemploLectura{
    public static void main(String [] args){
        InputStreamReader in = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(in);
        String texto;
        try{
            System.out.println("Introduce una cadena...");
            texto = br.readLine();
            System.out.println("Cadena escrita: "+texto);
            in.close();
        }catch(Exception e){ e.printStackTrace();}
    }
}
//EjemploLectura
```

Con el método **getOutputStream()** podemos lanzar el programa anterior y enviar datos a la entrada estándar del programa *EjemploLectura.java*. Por ejemplo, si queremos enviar la cadena “Hola Manuel”. ¿Cómo lo harías?

4.- Crea un programa Java que visualice 5 veces la cadena que se le envía desde los argumentos de *main()*. Si no se le envía ninguna cadena que muestre un mensaje indicándolo y que finalice el programa con *System.exit(1)*.

A continuación, crea un segundo programa Java que introduzca por teclado una cadena y ejecute el programa anterior para visualizar 5 veces esa cadena.