

EJERCICIOS PROCESOS

```
package paquete1;

import java.io.IOException;

public class Procesos {
    public static void main(String[] args) throws IOException, InterruptedException {

        //Ejercicio1:
        //Con la clase Runtime, que es la clase del sistema que te permite
        //acceder al entorno de ejecución java. Incluyendo la capacidad
        //de ejecutar comandos del sistema operativo
        //consigue lanzar un proceso que arranque el notepad en el que se
visualice
        //lo escrito en otro archivo llamado saludo.txt ubicado en la
        //raíz del proyecto que contiene el texto hola
        //comprueba si está vivo el proceso antes y después
        String[] param = {"notepad.exe", ".\\saludo.txt"}; //dir relativa
        Process p = Runtime.getRuntime().exec(param);
        System.out.println(p.isAlive()); //comprueba si está vivo el proceso
        p.waitFor(); //espera a que finalice el proceso (Ej. cuando le damos a
cerrar)
        p.destroy(); //destruye el proceso
        System.out.println(p.isAlive()); //comprueba si está vivo el proceso
    }
}
```

```

package paquete1;

import java.io.IOException;

public class Procesos {
    public static void main(String[] args) throws IOException, InterruptedException {

        //Ejercicio2:
        //Con la clase ProcessBuilder que es otra manera de ejecutar
        //procesos arranca el paint, el notepad y el chrome.
        ProcessBuilder pb1 = new ProcessBuilder("mspaint.exe");
        pb1.start();
        ProcessBuilder pb2 = new ProcessBuilder("notepad.exe");
        pb2.start();
        ProcessBuilder pb3 = new ProcessBuilder("\"C:\\Program
Files\\Google\\Chrome\\Application\\chrome.exe\""); //para conocer la ruta, ponerse
sobre el ejecutable->propiedades
        pb3.start();
    }
}

```

```

package paquete1;

import java.io.IOException;

public class Procesos {
    public static void main(String[] args) throws IOException, InterruptedException {

        //Ejercicio3: EN ALGUNOS SO NO FUNCIONÓ BIEN
        //Con ProcessBuilder arranca el paint, cuando lo cierras
        //arranca el notepad y cuando lo cierras, arranca el chrome.
        //Ayúdate de la clase Process
        ProcessBuilder pb1 = new ProcessBuilder("mspaint.exe");
        Process p1 = pb1.start();
        p1.waitFor();
        ProcessBuilder pb2 = new ProcessBuilder("notepad.exe");
        Process p2=pb2.start();
        p2.waitFor();
        ProcessBuilder pb3 = new ProcessBuilder("\"C:\\Program
Files\\Google\\Chrome\\Application\\chrome.exe\""); //para conocer la ruta, ponerse
sobre el ejecutable->propiedades
        Process p3 = pb3.start();
        p3.waitFor();
    }
}

```

Ejercicio 4

```

public class Principal {
    public static void main(String[] args) throws Exception{
        //Ejercicio 4
        //Crea un proceso que lance el Chrome con la clase Runtime. Mira su pid y
muestra su información (info)
        //comprueba si está vivo antes y después de destruir el proceso
        Process p = Runtime.getRuntime().exec("\"C:\\Program
Files\\Google\\Chrome\\Application\\chrome.exe\"");
        System.out.println(p.pid());
        System.out.println(p.info());
        System.out.println(p.isAlive());
        p.waitFor();//nos da el resultado del proceso cuando termina.
        // 0 es correcto. Distinto de cero no es correcto.
        //espera a que termine el proceso para poder continuar el

```

programa

```

        p.destroy();
        System.out.println(p.isAlive());
    }
}

```

EJERCICIO5

```

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Principal {
    public static void main(String[] args) throws Exception{
        //Ejercicio 5
        //Crea un proceso que muestre por pantalla la salida de hacer un ping a
        google.com
        //con ProcessBuilder
        ProcessBuilder pb = new ProcessBuilder();
        pb.command("ping","google.com"); //construcción del comando
        //si quisieras las tres primeras líneas:pb.command("ping","-
        c","3","google.com");
        Process p = pb.start();

        BufferedReader br = new BufferedReader(new
        InputStreamReader(p.getInputStream()));//capturamos la salida
        String linea;
        while ((linea=br.readLine())!= null) {
            System.out.println(linea);
        }

        int codSalida = p.waitFor();
        System.out.println(codSalida);
        p.destroy();//destruye el proceso si no acabó
    }
}

```

```

package paquete1;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Procesos {
    public static void main(String[] args) throws Exception{
        //Ejercicio 6
        //Crea un proceso que muestre por pantalla la salida de hacer un ping a
        google.com mediante Runtime
        Runtime r = Runtime.getRuntime();
        Process p = r.exec("ping google.com"); //construcción del comando

        BufferedReader br = new BufferedReader(new
        InputStreamReader(p.getInputStream()));//capturamos la salida
        String linea;
        while ((linea=br.readLine())!= null) {
            System.out.println(linea);
        }

        int codSalida = p.waitFor();
        System.out.println(codSalida);
        p.destroy();//destruye el proceso si no acabó
    }
}

```

```

package paquete1;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class Procesos {
    public static void main(String[] args) throws IOException, InterruptedException {

        //Ejercicio 7
        //Crea un proceso con ProcessBuilder que muestre por pantalla
        //los directorios o archivos de C:\\
        ProcessBuilder pb = new ProcessBuilder();
        pb.command("cmd", "/C", "dir", "c:\\"); //construcción del comando
        // C obliga a que termine el proceso
        //esto se podía condensar en ProcessBuilder pb = new
        ProcessBuilder("cmd", "/C", "dir", "c:\\");
        Process p = pb.start();

        BufferedReader br = new BufferedReader(new
        InputStreamReader(p.getInputStream())); //capturamos la salida
        //si en vez de getInputStream pones getErrorStream verías solo los
        errores, por ejemplo al poner dir1 en vez de dir
        String linea=null;
        while ((linea=br.readLine())!= null) {
            System.out.println(linea);
        }

        //otra forma sería ir letra a letra en vez de línea a línea
        /*InputStream is = p.getInputStream();
        int letra =0;
        while ((letra=is.read())!=-1) {
            System.err.print((char)letra);
        }*/

        int codSalida = p.waitFor();
        System.out.println(codSalida);
        p.destroy(); //destruye el proceso si no acabó
    }
}

```

```

package paquete1;

import java.io.File;
import java.io.IOException;

public class Procesos {
    public static void main(String[] args) throws InterruptedException, IOException {

        //Ejercicio 8
        //Crea un proceso con ProcessBuilder que guarde en un archivo
        //los directorios o archivos de C:\\
        ProcessBuilder pb = new ProcessBuilder();
        pb.command("cmd", "/C", "dir", "c:\\");
        pb.redirectOutput(new File("resultado.txt")); //este archivo te lo crea en el
        directorio raíz del proyecto
        Process p = pb.start();
    }
}

```

```
        int codSalida = p.waitFor();  
        System.out.println(codSalida);  
        p.destroy();//destruye el proceso si no acabó  
    }  
}
```