

//ejercicio 1 Realiza un programa en Java que ejecute el comando ipconfig a través de la //consola del sistema operativo y muestre su resultado por pantalla.

```
private static void ipconfig() {
    Runtime runtime = Runtime.getRuntime();
    Process process = null;
    String com = "cmd /c ipconfig";//llamamos al cmd para que ejecute el comando ipconfig
    System.out.println("ejecutamos el proceso hijo");
    try {
        process = runtime.exec(com); //ejecutamos el comando
        InputStream is = process.getInputStream(); //obtenemos la salida del comando
        BufferedReader br = new BufferedReader(new InputStreamReader(is)); //leemos la
        salida
        String line;
        while ((line = br.readLine()) != null) { //mientras haya lineas que leer las mostramos
            System.out.println(line);
        }
        br.close(); //cerramos el buffer
    } catch (Exception e) {
        System.err.println("Error:" + e.getMessage());
        System.exit(-1);
    }
}
```

//ejercicio 2 Realiza un programa en Java que abra la calculadora de Windows y muestre el //mensaje “La aplicación se ha cerrado con éxito.” al cerrar la aplicación ejecutada.

```
public static void calculator() {
    Runtime runtime = Runtime.getRuntime();
    Process process = null;
    String com = "calc";//llamamos al cmd para que el proceso hijo ejecute el comando calc
    try {
        process = runtime.exec(com); //ejecutamos el comando
        // esperamos a que el proceso se termine y mostramos el mensaje
        if (process.waitFor() == 0) System.out.println("La aplicación se ha cerrado con éxito.");
    } catch (Exception e) {
        System.err.println("Error:" + e.getMessage());
        System.exit(-1);
    }
}
```

//ejercicio 3 Realiza un programa en Java que admita como parámetro de entrada el comando

//a ejecutar en la consola del sistema operativo y muestre en pantalla el resultado.

```
public static void command(String com) {
    if (com == null) { //controlamos que se hayan introducido parametros
        System.out.println("Error: No se ha introducido ningún comando");
        System.exit(-1);
    }
    Runtime runtime = Runtime.getRuntime();
}
```

```

Process process;
try {
    process = runtime.exec(com); //ejecutamos el comando
    InputStream is = process.getInputStream(); //obtenemos la salida
    BufferedReader br = new BufferedReader(new InputStreamReader(is)); //leemos la
salida
    String line;
    while ((line = br.readLine()) != null) { //mientras haya lineas que leer las mostramos
        System.out.println(line);
    }
    br.close();
} catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
}
}

```

//ejercicio 4 . Realiza un programa en Java que admita como parámetro de entrada la ruta de

//un archivo ejecutable, ejecute dicha aplicación y muestre por pantalla el mensaje

//“Aplicación finalizada.” al finalizar la aplicación ejecutada.

```

public static void ejecuta(String com) {
    if (com == null) { //controlamos que se hayan introducido parametros
        System.out.println("Error: No se ha introducido ningún comando");
        System.exit(-1);
    }
    Runtime runtime = Runtime.getRuntime();
    try {
        Process process = runtime.exec(com); //ejecutamos el comando
        if (process.waitFor() == 0) System.out.println("Aplicación finalizada."); //esperamos a
que termine y mostramos el mensaje
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
}
}

```

//ejercicio 5 Modificar el programa anterior para que admita como parámetros de entrada más

//de una aplicación y espere a que todas terminen para mostrar el mensaje por

//pantalla “Aplicaciones finalizadas.”.

```

public static void ejecuta2(String[] com) {
    int q = 0;
    if (com == null) { //controlamos que se hayan introducido parametros
        System.out.println("Error: No se ha introducido ningún parametro");
        System.exit(-1);
    }
    else if (com.length > 10) { //controlamos que no se hayan introducido mas de 10
parametros
        System.out.println("Error: demasiados parametros");
    }
}

```

```

        System.exit(-1);
    } else {
        Runtime runtime = Runtime.getRuntime();
        try {
            for (int i = 0; i < com.length; i++) {
                Process process = runtime.exec(com[i]); //ejecutamos el comando
                q += process.waitFor(); //esperamos a que termine y sumamos el resultado
            }
            if (q == 0) System.out.println("Aplicaciones finalizadas correctamente.");
            //si la suma de los resultados es 0 mostramos el mensaje
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

//ejercicio 6 Realizar un programa en Java que obtenga la dirección IP del ordenador y la  
 //muestre por pantalla haciendo uso de lo visto en esta unidad (ejecutando el  
 //comando ipconfig).

```

private static void ip() {
    Runtime runtime = Runtime.getRuntime();
    Process process = null;
    String com = "cmd c/ ipconfig";
    try {
        process = runtime.exec(com); //ejecutamos el comando
        InputStream is = process.getInputStream(); //obtenemos la salida
        BufferedReader br = new BufferedReader(new InputStreamReader(is)); //leemos la
        salida
        String line;
        while ((line = br.readLine()) != null) { //mientras haya lineas que leer las mostramos
            //si en la linea coincide en una posicion indeterminada la secuencia IPv4 luego otra
            //serie de caracteres indeterminados seguidos de la secuencia 192 en una posicion
            indeterminada
            //entonces mostramos esa linea por pantalla
            if (line.matches(".*IPv4.*192.*")) {
                System.out.println(line);
            }
        }
        br.close();
    } catch (Exception e) {
        System.err.println("Error:" + e.getMessage());
        System.exit(-1);
    }
}

```

```

//ejercicio 7 Escribe un programa Aleatorios que haga lo siguiente:
//I. Cree un proceso hijo que está encargado de generar números aleatorios.
// Para su creación se puede usar el ejemplo dejado en el campus. Este
// proceso hijo escribirá en su salida estándar un número aleatorio del 0 al 10
// cada vez que reciba una petición de ejecución por parte del padre. Se creará
// el ejecutable .jar y será invocado desde el proceso padre
//II. El proceso padre lee líneas de la entrada estándar y por cada línea que lea
// solicitará al hijo que le envíe un número aleatorio, lo leerá y lo imprimirá en
// pantalla
//III. Cuando el proceso padre reciba la palabra "fin", finalizará la ejecución del
// hijo y procederá a finalizar su ejecución
private static void aleatorios() {
    InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader bw = new BufferedReader(isr);
    String entrada = null;
    Runtime r = Runtime.getRuntime();
    Process p = null;
    String com = "java -jar
C:\\Users\\usuario\\Desktop\\DAM\\PSP\\U1\\Practica1.1Ej7\\Aleatorio.jar";
    try {
        p = r.exec(com); //ejecutamos el comando
        InputStream is = p.getInputStream(); //obtenemos la salida hacia el proceso hijo
        BufferedReader br = new BufferedReader(new InputStreamReader(is)); //leemos la
salida del proceso hijo
        OutputStream os = p.getOutputStream(); //entrada de datos hacia el proceso hijo
        System.out.println("Introduce texto (fin para terminar):");
        while((entrada = bw.readLine()) != null && !entrada.equalsIgnoreCase("fin")) {
            //mientras escribamos algo y no sea fin se lo enviamos al proceso hijo
            entrada = entrada + "\n";
            os.write(entrada.getBytes()); //escribimos en la salida del proceso hijo lo que hemos
escrito
            os.flush(); //limpiamos el buffer
            System.out.println("Numero aleatorio: " + br.readLine() ); //mostramos el numero
aleatorio que nos envia el proceso hijo
        }
        entrada += "\n";
        os.write(entrada.getBytes()); //enviamos fin al proceso hijo para que termine
        os.flush();
        os.close();
        int exitVal = p.waitFor();
        System.out.println("Valor de salida: " + exitVal);
    }
    catch (IOException ex){
        System.out.println("error: "+ex.getMessage());
    }catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```