

PROGRAMACIÓN DE SERVICIOS Y PROCESOS



PROGRAMACIÓN DE
SERVICIOS Y PROCESOS

15/12/2019

DAVID BORREGO ASENCIO

ÍNDICE

CÓDIGO JAVA.....	1
CAPTURAS DE PANTALLA	11
BIBLIOGRAFÍA	15

CÓDIGO JAVA

```

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import javax.swing.border.*;

public class Procesos extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField textField;
    private int retorno;
    private Runtime runtime;
    private Process process;
    private JTextArea areadetexto;
    private File file;
    private FileReader filereader;
    private BufferedReader bufferedreader;
    private String linea_texto;
    private List list;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Procesos frame = new Procesos();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public Procesos()
    {

        file = new File("C:" + File.separator + "temp" +
File.separator + "temp.txt");
        file.mkdir(); //CREAMOS EL DIRECTORIO CADA VEZ QUE
INICIAMOS EL PROGRAMA

        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 725, 781);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(new BorderLayout(0, 0));

        JPanel panel = new JPanel();
        panel.setBackground(new Color(0, 0, 205));
        contentPane.add(panel, BorderLayout.CENTER);
    }
}

```

```

panel.setLayout(null);

textField = new JTextField();
textField.setForeground(new Color(0, 255, 0));
textField.setBackground(new Color(0, 0, 0));
textField.setFont(new Font("Times New Roman", Font.PLAIN,
17));

textField.setBounds(20, 44, 243, 24);
panel.add(textField);
textField.setColumns(10);

JButton btn_Bloc_De_Notas = new JButton("Bloc de Notas");
btn_Bloc_De_Notas.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/blocde
notas.png")));
btn_Bloc_De_Notas.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try
        {
            Runtime runtime =
Runtime.getRuntime();//CREAMOS EL RUNTIME
            Process process = runtime.exec("cmd /c
notepad");//EJECUTAMOS EN EL CMD
            System.out.println("-----
-----");

            System.out.println("cmd /c notepad");
            System.out.println("-----
-----");

            btn_Bloc_De_Notas.setEnabled(false);

//APAGAMOS EL BOTON

            retorno = process.waitFor(); //RECOGEMOS
EL VALOR DE RETORNO DEL PROGRAMA, SI SE CIERRA POR CONVENCION ES UN 0

            if(retorno==0)
            {
                btn_Bloc_De_Notas.setEnabled(true);

//ENCENDEMOS EL BOTON CUANDO ACABA EL PROCESO
            }

        } catch (IOException e1)
        {
            e1.printStackTrace();
        } catch (InterruptedException e1)
        {
            e1.printStackTrace();
        }
    }
});

btn_Bloc_De_Notas.setForeground(new Color(255, 255, 255));
btn_Bloc_De_Notas.setBackground(new Color(220, 20, 60));
btn_Bloc_De_Notas.setBounds(509, 98, 137, 46);
panel.add(btn_Bloc_De_Notas);

JButton btn_Paint = new JButton("Paint");
btn_Paint.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/paint.
png")));
btn_Paint.setForeground(new Color(255, 255, 255));

```

```

        btn_Paint.setBackground(new Color(220, 20, 60));
        btn_Paint.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {

                try
                {
                    Runtime runtime =
Runtime.getRuntime();//CREAMOS EL RUNTIME
                    Process process = runtime.exec("cmd /c
mspaint");//EJECUTAMOS EN EL CMD
                    System.out.println("-----
-----");
                    System.out.println("cmd /c mspaint");
                    System.out.println("-----
-----");
                    btn_Bloc_De_Notas.setEnabled(false);
//APAGAMOS EL BOTON

                    retorno = process.waitFor(); //RECOGEMOS
EL VALOR DE RETORNO DEL PROGRAMA, SI SE CIERRA POR CONVENCION ES UN 0

                    if( retorno == 0)
                    {
                        btn_Bloc_De_Notas.setEnabled(true);
                    }

                }catch (IOException e1)
                {
                    e1.printStackTrace();
                } catch (InterruptedException e1)
                {
                    e1.printStackTrace();
                }
            }
        });
        btn_Paint.setBounds(509, 173, 137, 46);
        panel.add(btn_Paint);

        JButton btn_Gestion = new JButton("Gesti\u00F3n");
        btn_Gestion.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/gestio
n.png")));
        btn_Gestion.setBackground(new Color(220, 20, 60));
        btn_Gestion.setForeground(new Color(255, 255, 255));
        btn_Gestion.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                try
                {
                    Runtime runtime =
Runtime.getRuntime();//CREAMOS EL RUNTIME
                    Process process = runtime.exec("cmd /c
C:\\\\Users\\\\david\\\\Desktop\\\\Policlinica.jar");//EJECUTAMOS EN
EL CMD
                    System.out.println("-----
-----");
                    System.out.println("cmd /c
C:\\Users\\david\\Desktop\\Policlinica.jar");

```

```

        System.out.println("-----
        -----");
        btn_Gestion.setEnabled(false); //APAGAMOS
        EL BOTON

        retorno = process.waitFor(); //RECOGEMOS
        EL VALOR DE RETORNO DEL PROGRAMA, SI SE CIERRA POR CONVENCION ES UN 0

        if( retorno == 0)
        {
            btn_Gestion.setEnabled(true);
        }

        } catch (IOException e1)
        {
            e1.printStackTrace();
        } catch (InterruptedException e1)
        {
            e1.printStackTrace();
        }

    }

});
btn_Gestion.setBounds(509, 245, 137, 46);
panel.add(btn_Gestion);

JButton btn_Juego = new JButton("Juego");
btn_Juego.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try
        {
            Runtime runtime =
Runtime.getRuntime();//CREAMOS EL RUNTIME
            Process process = runtime.exec("cmd /c
C:\\Users\\david\\Desktop\\Snake.jar");//EJECUTAMOS EN EL CMD
            System.out.println("-----
            -----");
            System.out.println("cmd /c
C:\\Users\\david\\Desktop\\Snake.jar");
            System.out.println("-----
            -----");
            btn_Juego.setEnabled(false); //APAGAMOS
            EL BOTON

            retorno = process.waitFor(); //RECOGEMOS
            EL VALOR DE RETORNO DEL PROGRAMA, SI SE CIERRA POR CONVENCION ES UN 0

            if( retorno == 0)
            {
                btn_Juego.setEnabled(true);
            }

            } catch (IOException e1)
            {
                e1.printStackTrace();
            } catch (InterruptedException e1)

```

```

        {
            e1.printStackTrace();
        }
    });
    btn_Juego.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/snake.
png")));
    btn_Juego.setBackground(new Color(220, 20, 60));
    btn_Juego.setForeground(new Color(255, 255, 255));
    btn_Juego.setBounds(509, 314, 137, 46);
    panel.add(btn_Juego);

    JButton btn_Terminar = new JButton("Terminar");
    btn_Terminar.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/elimin
ar.png")));
    btn_Terminar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

            //-----ELIMINA EL
PROCESO SELECCIONADO EN EL LIST DE NUESTRA INTERFAZ Y EJECUTA EL
COMANDO EN EL CMD-----

            String matriz = list.getSelectedItem();
            char [] cadena_div = matriz.toCharArray();
            String pid_proceso = "";
            boolean llave = true;
            boolean llave2 = true;
            int i = 0;

            while(i < cadena_div.length && llave2 == true)
//RECOGE LA PRIMERA SECUENCIA DE NUMEROS EN EL STRING, CORRESPONDIENTE
AL PID DEL PROCESO
            {

                if(Character.isDigit(cadena_div[i]))
                {
                    pid_proceso += cadena_div[i];
                    llave = false;
                }

                if(llave == false &&
!Character.isDigit(cadena_div[i]))
                {
                    llave2 = false;
                }
                i++;
            }
            System.out.println("-----
-----");
            System.out.println("PID del proceso: " +
pid_proceso);

            try
            {

                runtime = Runtime.getRuntime();//CREAMOS
EL RUNTIME

```



```

        process = runtime.exec("cmd /c taskkill
/pid " + pid_proceso); //EJECUTAMOS EN EL CMD
        System.out.println("cmd /c taskkill /pid
" + pid_proceso);

        } catch (IOException e1) {
            System.out.println(e1);
        }
        System.out.println("-----
-----");

        list.remove(list.getSelectedItem()); //ELIMINA
DEL LIST EL ELEMENTO SELECCIONADO
//-----
-----C

    }
});
btn_Terminar.setBackground(new Color(220, 20, 60));
btn_Terminar.setForeground(new Color(255, 255, 255));
btn_Terminar.setBounds(509, 649, 137, 41);
panel.add(btn_Terminar);
areadetexto = new JTextArea();
areadetexto.setBackground(new Color(0, 0, 0));
areadetexto.setForeground(new Color(124, 252, 0));
areadetexto.setFont(new Font("Times New Roman", Font.PLAIN,
13));

JButton btn_Ejecutar = new JButton("Ejecutar");
btn_Ejecutar.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/ejecut
ar.png")));
btn_Ejecutar.setForeground(new Color(255, 255, 255));
btn_Ejecutar.setBackground(new Color(0, 204, 0));
btn_Ejecutar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try
        {

            runtime = Runtime.getRuntime(); //CREAMOS
EL RUNTIME

            process = runtime.exec("cmd /c " +
textField.getText() + " > C:" + File.separator + "temp" + File.separator
+"temp.txt"); //EJECUTAMOS EN EL CMD
            System.out.println("cmd /c " +
textField.getText() + " > C:" + File.separator + "temp" + File.separator
+"temp.txt");

            Thread.sleep(500); //DORMIMOS PARA QUE DE
TIEMPO A CARGAR BIEN LUEGO EL AREADETEXTO CON LA INFORMACIÓN DEL TXT

        } catch (IOException e1) {
            System.out.println(e1);
        } catch (InterruptedException e1)
        {
            System.out.println("Error en la función
sleep");

```

```

    }

    try
    {
        filereader = new FileReader(file);
        bufferedreader = new
BufferedReader(filereader);

        } catch (FileNotFoundException e2)
        {
            System.out.println("Error con  FileReader
y  BufferedReader");
        }

        try {

            linea_texto = bufferedreader.readLine();

            while(linea_texto != null)
            {

                areadetexto.setText(areadetexto.getText() + linea_texto + "
\n");

                linea_texto =
bufferedreader.readLine(); //AÑADIMOS UN SALTO DE CARRO PARA LA
SIGUIENTE LINEA

            }
        } catch (Exception e1) {

            System.err.println("Error");
            e1.printStackTrace();

        }

    }

    btn_Ejecutar.setBounds(286, 44, 112, 25);
    panel.add(btn_Ejecutar);

    JScrollPane scrollPane = new JScrollPane(areadetexto);
    scrollPane.setBounds(20, 91, 459, 269);
    panel.add(scrollPane);

    JSeparator separator = new JSeparator();
    separator.setBounds(20, 379, 658, 2);
    panel.add(separator);

    list = new List();
    list.setFont(new Font("Times New Roman", Font.PLAIN, 13));
    list.setBackground(Color.BLACK);
    list.setForeground(new Color(0, 255, 0));
    list.setBounds(20, 421, 459, 269);
    panel.add(list);

    JButton btnNewButton = new JButton("Refrescar");
    btnNewButton.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/actual
izar.png")));
    btnNewButton.setBackground(new Color(220, 20, 60));
    btnNewButton.setForeground(Color.WHITE);
    btnNewButton.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent arg0) {

            list.removeAll();//BORRA EL LIST PARA VOLVER A
CARGARLO

            //-----//RELLENAMOS LA
TERMINAL DE PROCESOS CON UN TASKLIST-----

            try
            {

                runtime = Runtime.getRuntime();//CREAMOS
EL RUNTIME
                process = runtime.exec("cmd /c tasklist"
+ " > C:" + File.separator + "temp" + File.separator
+ "temp.txt");//EJECUTAMOS EN EL CMD
                System.out.println("-----
-----");
                System.out.println("cmd /c tasklist" + "
> C:" + File.separator + "temp" + File.separator + "temp.txt");
                System.out.println("-----
-----");

                Thread.sleep(500); //DORMIMOS PARA QUE DE
TIEMPO A CARGAR BIEN LUEGO EL AREADETEXTO CON LA INFORMACIÓN DEL TXT

            } catch (IOException e1) {
                System.out.println(e1);
            } catch (InterruptedException e1)
            {
                System.out.println("Error en el SLEEP del
procesador");
            }

            try
            {
                filereader = new FileReader(file);
                bufferedreader = new
BufferedReader(filereader);

            } catch (FileNotFoundException e2)
            {
                System.out.println("Error con FileReader
y BufferedReader");
            }

            try {

                linea_texto = bufferedreader.readLine();

                while(linea_texto != null)
                {
                    list.add(linea_texto + " \n");
                    linea_texto =
bufferedreader.readLine(); //AÑADIMOS UN SALTO DE CARRO PARA LA
SIGUIENTE LINEA

                }
            }
        }
    }
}

```

```

        } catch (Exception e1) {

            System.err.println("Error 404");
            e1.printStackTrace();
        }

//-----
-----

    }
});
btnNewButton.setBounds(509, 556, 137, 41);
panel.add(btnNewButton);

JLabel lblProcesosDelSistema = new JLabel("PROCESOS DEL
SISTEMA");
lblProcesosDelSistema.setForeground(new Color(0, 255,
255));
lblProcesosDelSistema.setFont(new Font("Times New Roman",
Font.BOLD, 20));
lblProcesosDelSistema.setBounds(20, 392, 257, 24);
panel.add(lblProcesosDelSistema);

JLabel lblConsolaDeComandos = new JLabel("CONSOLA DE
COMANDOS");
lblConsolaDeComandos.setForeground(new Color(0, 255, 255));
lblConsolaDeComandos.setFont(new Font("Times New Roman",
Font.BOLD, 20));
lblConsolaDeComandos.setBounds(20, 11, 257, 24);
panel.add(lblConsolaDeComandos);

JLabel lblNewLabel = new JLabel("");
lblNewLabel.setIcon(new
ImageIcon(Procesos.class.getResource("/es/grupostudium/Procesos/consol
acomandos.png")));
lblNewLabel.setBounds(529, 404, 112, 125);
panel.add(lblNewLabel);

JLabel lblDesarrolladoPorDavid = new JLabel("Desarrollado
por David Borrego Asencio");
lblDesarrolladoPorDavid.setForeground(new Color(0, 0, 0));
lblDesarrolladoPorDavid.setBounds(457, 718, 252, 24);
panel.add(lblDesarrolladoPorDavid);

//-----//RELLENAMOS LA TERMINAL DE
PROCESOS CON UN TASKLIST-----

try
{

    runtime = Runtime.getRuntime();//CREAMOS EL RUNTIME
    process = runtime.exec("cmd /c tasklist" + " >
C:"+File.separator + "temp" + File.separator + "temp.txt");//EJECUTAMOS
EN EL CMD

```

```

        System.out.println("-----
--");
        System.out.println("cmd /c tasklist" + " >
C:"+File.separator + "temp" + File.separator + "temp.txt");
        System.out.println("-----
--");
        Thread.sleep(500); //DORMIMOS PARA QUE DE TIEMPO A
CARGAR BIEN LUEGO EL AREADETEXTO CON LA INFORMACIÓN DEL TXT

    } catch (IOException e1) {
        System.out.println(e1);
    } catch (InterruptedException e1)
    {
        System.out.println("Error en el SLEEP del
procesador");
    }

    try
    {
        filereader = new FileReader(file);
        bufferedreader = new BufferedReader(filereader);

    } catch (FileNotFoundException e2)
    {
        System.out.println("Error con FileReader y
BufferedReader");
    }

    try {

        linea_texto = bufferedreader.readLine();

        while(linea_texto != null)
        {

            //System.out.println(linea_texto + " \n");
            list.add(linea_texto + " \n");
            //areadetexto2.append(linea_texto + " \n");
            linea_texto = bufferedreader.readLine();
//AÑADIMOS UN SALTO DE CARRO PARA LA SIGUIENTE LINEA

        }

    } catch (Exception e1) {

        System.err.println("Error 404");
        e1.printStackTrace();
    }

    //-----
-----

}
}

```

CAPTURAS DE PANTALLA

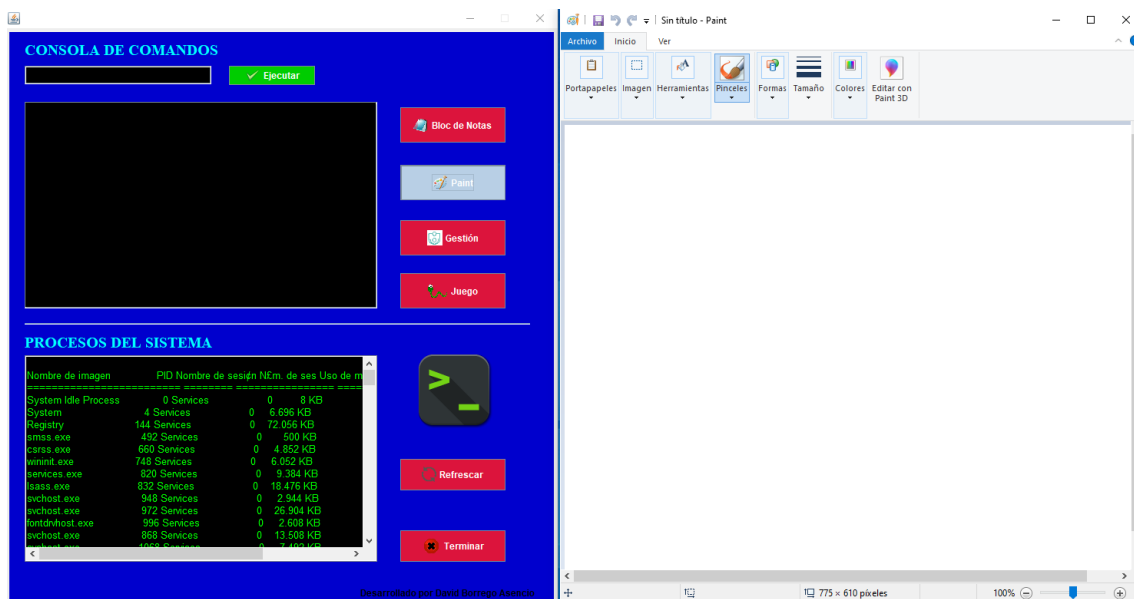
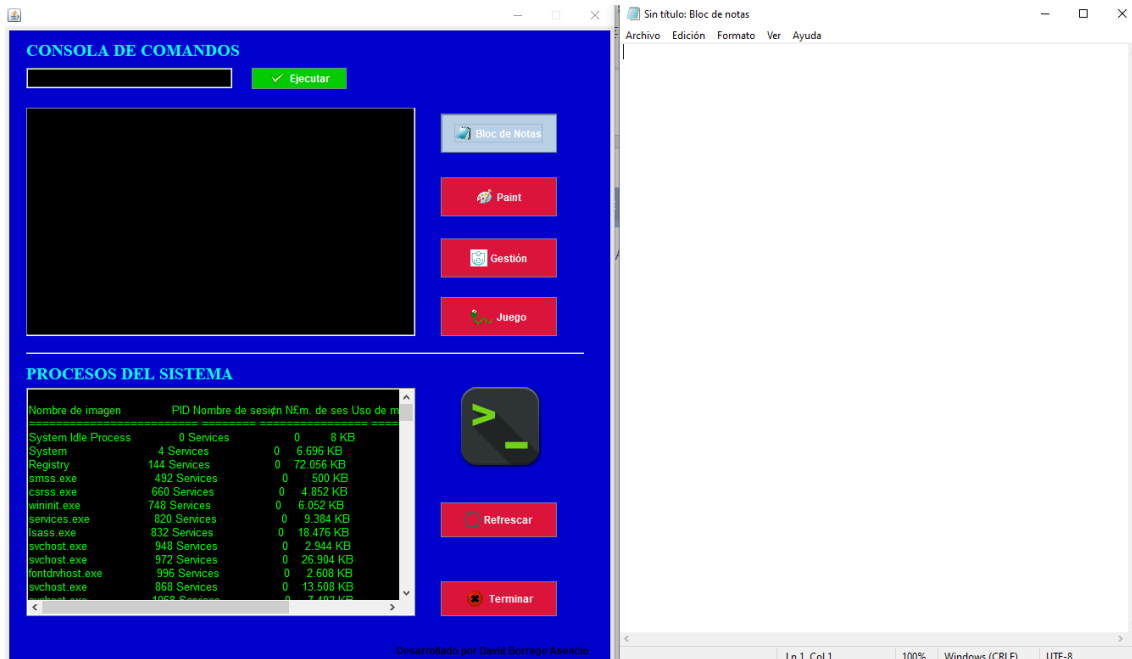
Al abrir el programa nos encontraremos con esta pantalla única del programa. En ella encontramos dos áreas bien diferenciadas: consola de comandos y procesos del sistema.

En la consola de comandos podemos ejecutar los comandos que necesitemos, tan solo tenemos que escribir la orden a en el “JTextField” y hacer clic en “Ejecutar”.



The screenshot shows a Windows 10 desktop environment with a blue background. On the left side, there are two command prompt windows. The top window, titled 'CONSOLA DE COMANDOS', shows the output of the 'dir' command, displaying file information for the 'E:\Archivos de Programa\JAVA 2019\Procesos' directory. The bottom window, titled 'PROCESOS DEL SISTEMA', shows a list of running processes with columns for 'Nombre de imagen', 'PID', 'Nombre de sesión', 'Núm. de ses', and 'Uso de m'. On the right side of the desktop, there are several application icons: 'Bloc de Notas' (Notepad), 'Paint', 'Gestión' (Management), 'Juego' (Game), 'Refrescar' (Refresh), and 'Terminar' (End). The taskbar at the bottom shows the Start button and several pinned applications, including the Command Prompt, File Explorer, and the 'Juego' application.

A la derecha de esta área podemos encontrar cuatro botones: Bloc de Notas, Paint, Gestión y Juego. Al hacer clic se abrirá el programa correspondiente en nuestro sistema operativo, quedando bloqueado el botón hasta que se cierre el programa ejecutado.



Por otro lado, tenemos el área de Procesos del Sistema. En esta área encontramos una lista de procesos que se están ejecutando en nuestro sistema. Para parar la ejecución de un proceso solo tenemos que seleccionar el proceso que queremos eliminar y hacer clic en “Terminar”, eliminando así el proceso de la lista.

El botón de “Refrescar” actualiza la lista de procesos que se están ejecutando, en caso de que abramos un nuevo proceso y queramos que se vea en la lista.



BIBLIOGRAFÍA

- Apuntes PSP. Tema 1. Grupo Studium. 2019.
- Google Imágenes.