

```

public class Main {
    public static void main(String[] args) {
        //ipconfig();
        //calculator();
        //command(args[0]);
        //ejecuta(args[0]);
        //ejecuta2(args);
        //ip();
    }
}

```

//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 = "ipconfig";//llamamos al cmd para que ejecute el comando
    ipconfig
    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 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){
        System.out.println("Error: No se ha introducido ningún comando");
        System.exit(-1);
    }
    Runtime runtime = Runtime.getRuntime();
    Process process;
    try{
        process = runtime.exec(com);
        InputStream is = process.getInputStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(is));
        String line;
        while ((line = br.readLine()) != null) {
            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){

```

```

        System.out.println("Error: No se ha introducido ningún comando");
        System.exit(-1);
    }
    Runtime runtime = Runtime.getRuntime();
    try{
        Process process = runtime.exec(com);
        if(process.waitFor()==0) System.out.println("Aplicación finalizada.");
    } 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){
    if(com==null){
        System.out.println("Error: No se ha introducido ningún parametro");
        System.exit(-1);
    }
    if(com.length>10){
        System.out.println("Error: demasiados parametros");
        System.exit(-1);
    }
    Runtime runtime = Runtime.getRuntime();
    try{
        for (int i = 0; i < com.length; i++){
            Process process = runtime.exec(com[i]);
            process.waitFor();
        }
        System.out.println("Aplicaciones finalizadas.");
    } 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 = "ipconfig";
try {
    process = runtime.exec(com);
    InputStream is = process.getInputStream();
    BufferedReader br = new BufferedReader(new InputStreamReader(is));
    String line;
    while ((line = br.readLine()) != null) {
        //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);
}
}
}

```