

Ejercicios prácticos con ProcessBuilder

Ejercicio 1: Lanzador de programas básico

Crea un programa que pueda ejecutar diferentes comandos según una opción:

java

```
import java.io.*;
import java.util.*;

public class Lanzador {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("1. Ver archivos del directorio");
        System.out.println("2. Ver fecha y hora");
        System.out.println("3. Abrir calculadora");
        System.out.print("Elige opción: ");

        int opcion = sc.nextInt();

        try {
            ProcessBuilder pb = null;

            switch(opcion) {
                case 1:
                    pb = new ProcessBuilder("cmd.exe", "/c", "dir");
                    break;
                case 2:
                    pb = new ProcessBuilder("cmd.exe", "/c", "date /t && time /t");
                    break;
                case 3:
                    pb = new ProcessBuilder("calc.exe");
                    break;
                default:
                    System.out.println("Opción no válida");
                    return;
            }

            pb.inheritIO(); // Muestra salida en nuestra consola
```

```

        Process p = pb.start();
        p.waitFor();

    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Ejercicio 2: Contador de líneas

Programa que cuenta líneas de un fichero usando wc (Linux) o PowerShell (Windows):

java

```

import java.io.*;

public class ContadorLineas {
    public static void main(String[] args) throws IOException, InterruptedException {
        if (args.length != 1) {
            System.out.println("Uso: java ContadorLineas <fichero>");
            return;
        }

        File fichero = new File(args[0]);
        if (!fichero.exists()) {
            System.out.println("El fichero no existe");
            return;
        }

        // Para Windows (PowerShell)
        ProcessBuilder pb = new ProcessBuilder(
            "powershell.exe",
            "-Command",
            "Get-Content '" + fichero.getPath() + "' | Measure-Object -Line"
        );

        // Para Linux/Mac sería:
        // ProcessBuilder pb = new ProcessBuilder("wc", "-l", fichero.getPath());

        Process p = pb.start();
    }
}

```

```

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

        p.waitFor();
    }
}

```

✦ Ejercicio 3: Probador de conexión (ping)

```

java

import java.io.*;

public class TestConexion {
    public static void main(String[] args) {
        String[] hosts = {"google.com", "github.com", "localhost"};

        for (String host : hosts) {
            System.out.println("\nProbando conexión con: " + host);

            try {
                ProcessBuilder pb;

                if (System.getProperty("os.name").toLowerCase().contains("win")) {
                    // Windows
                    pb = new ProcessBuilder("ping", "-n", "3", host);
                } else {
                    // Linux/Mac
                    pb = new ProcessBuilder("ping", "-c", "3", host);
                }

                Process p = pb.start();

                // Leer salida línea por línea
                BufferedReader reader = new BufferedReader(

```

```

        new InputStreamReader(p.getInputStream())
    );

    String linea;
    while ((linea = reader.readLine()) != null) {
        System.out.println(linea);
    }

    int exitCode = p.waitFor();
    System.out.println("Código de salida: " + exitCode);

} catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
}
}
}
}
}

```

🔴 Ejercicio 4: Compilador y ejecutor Java automático

java

```
import java.io.*;
```

```
public class AutoCompilador {
    public static void main(String[] args) throws IOException, InterruptedException {
        String archivoJava = "HolaMundo.java";
```

```
// 1. Crear un fichero Java de ejemplo
```

```
try (PrintWriter pw = new PrintWriter(archivoJava)) {
    pw.println("public class HolaMundo {");
    pw.println("    public static void main(String[] args) {");
    pw.println("        System.out.println(\"¡Hola desde proceso hijo!\");");
    pw.println("        for(int i = 1; i <= 3; i++) {");
    pw.println("            System.out.println(\"Contador: \" + i);");
    pw.println("        }");
    pw.println("    }");
    pw.println("}");
}

```

```
System.out.println("1. Compilando " + archivoJava + "...");
```

```

// 2. Compilar
ProcessBuilder pbCompilar = new ProcessBuilder("javac", archivoJava);
Process pCompilar = pbCompilar.start();
pCompilar.waitFor();

if (pCompilar.exitValue() == 0) {
    System.out.println("✓ Compilación exitosa");

    // 3. Ejecutar
    System.out.println("\n2. Ejecutando HolaMundo...");
    ProcessBuilder pbEjecutar = new ProcessBuilder("java", "HolaMundo");
    pbEjecutar.inheritIO(); // Muestra salida directamente
    Process pEjecutar = pbEjecutar.start();
    pEjecutar.waitFor();

    System.out.println("\n✓ Proceso terminado con código: " + pEjecutar.exitValue());

} else {
    System.out.println("X Error en la compilación");
}

// 4. Limpiar
new File(archivoJava).delete();
new File("HolaMundo.class").delete();
}
}

```

Ejercicio 5: Sistema de backup simple

```

java

import java.io.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class BackupAutomatico {
    public static void main(String[] args) throws IOException, InterruptedException {
        // Crear nombre de backup con fecha
        String fecha =
        LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyyMMdd_HHmmss"));
    }
}

```

```

String archivoBackup = "backup_" + fecha + ".zip";

System.out.println("Creando backup: " + archivoBackup);

// Para Windows (usando 7-Zip o PowerShell)
// Si tienes 7-Zip:
ProcessBuilder pb = new ProcessBuilder(
    "7z.exe", "a", archivoBackup, "*.java", "*.txt"
);

// Alternativa con PowerShell (Windows):
// ProcessBuilder pb = new ProcessBuilder(
//     "powershell.exe",
//     "Compress-Archive",
//     "-Path", "*.java,*.txt",
//     "-DestinationPath", archivoBackup
// );

// Para Linux/Mac:
// ProcessBuilder pb = new ProcessBuilder(
//     "zip", archivoBackup, "*.java", "*.txt"
// );

pb.inheritIO();
Process p = pb.start();
int resultado = p.waitFor();

if (resultado == 0) {
    System.out.println("✓ Backup creado exitosamente");

    // Mostrar información del archivo creado
    ProcessBuilder pbInfo = new ProcessBuilder(
        "cmd.exe", "/c", "dir " + archivoBackup
    );
    pbInfo.inheritIO();
    pbInfo.start().waitFor();
} else {
    System.out.println("X Error creando backup");
}
}
}

```

📌 Ejercicio 6: Procesador de ficheros por lotes

java

```
import java.io.*;

public class ProcesadorLotes {
    public static void procesarFichero(String nombreFichero) throws IOException,
    InterruptedException {
        System.out.println("\n--- Procesando: " + nombreFichero + " ---");

        // Ejemplo: contar palabras (Windows PowerShell)
        ProcessBuilder pb = new ProcessBuilder(
            "powershell.exe",
            "-Command",
            "$texto = Get-Content '" + nombreFichero + "' -Raw; " +
            "$palabras = ($texto -split '\s+').Count; " +
            "Write-Host 'Palabras:' $palabras; " +
            "Write-Host 'Líneas:' (Get-Content '" + nombreFichero + "').Count"
        );

        Process p = pb.start();

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

        p.waitFor();
    }

    public static void main(String[] args) throws IOException, InterruptedException {
        // Lista de ficheros a procesar
        String[] ficheros = {"documento1.txt", "documento2.txt", "datos.csv"};

        // Crear algunos ficheros de ejemplo
        for (int i = 0; i < ficheros.length; i++) {
```

```

        try (PrintWriter pw = new PrintWriter(ficheros[i])) {
            pw.println("Este es el fichero número " + (i+1));
            pw.println("Contiene varias líneas de texto");
            pw.println("para probar el procesamiento por lotes");
        }
    }

    // Procesar cada fichero
    for (String fichero : ficheros) {
        procesarFichero(fichero);
    }

    // Limpiar
    for (String fichero : ficheros) {
        new File(fichero).delete();
    }
}

```

🔑 Conceptos clave que practicas:

ProcessBuilder.start() - Iniciar procesos

Process.waitFor() - Esperar a que termine

getInputStream() - Leer salida del proceso

inheritIO() - Redirigir entrada/salida

Manejo de diferentes SO (Windows/Linux)

Captura de errores y códigos de salida

¿Cuál te gustaría desarrollar más? Puedo darte el código completo con comentarios paso a paso.

y alguno mas facil tipo abrir block de notas

Ejercicios FÁCILES con ProcessBuilder

📁 Ejercicio 1: Abrir Bloc de Notas (Notepad)

java


```

import java.io.*;

public class AbrirNotepad {
    public static void main(String[] args) {
        try {
            System.out.println("Abriendo Bloc de notas...");

            // Para Windows
            ProcessBuilder pb = new ProcessBuilder("notepad.exe");

            Process proceso = pb.start();
            System.out.println("Bloc de notas abierto. PID: " + proceso.pid());

            // Esperar a que el usuario cierre Notepad
            int codigoSalida = proceso.waitFor();
            System.out.println("Bloc de notas cerrado. Código: " + codigoSalida);

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

Ejercicio 2: Abrir Bloc de Notas con un archivo

```

java

import java.io.*;

public class AbrirNotepadConArchivo {
    public static void main(String[] args) throws IOException, InterruptedException {
        // Crear un archivo de texto primero
        File archivo = new File("miNota.txt");

        try (PrintWriter pw = new PrintWriter(archivo)) {
            pw.println("Esta es una nota creada desde Java");
            pw.println("Fecha: " + new java.util.Date());
            pw.println("-----");
            pw.println("¡Hola Mundo!");
        }
    }
}

```

```

System.out.println("Archivo creado: " + archivo.getAbsolutePath());

// Abrir Notepad con ese archivo
ProcessBuilder pb = new ProcessBuilder("notepad.exe", archivo.getPath());
Process p = pb.start();

System.out.println("Notepad abierto. Cierra Notepad para continuar...");
p.waitFor();

System.out.println("Notepad cerrado");

// Opcional: borrar el archivo
// archivo.delete();
}
}

```

Ejercicio 3: Menú para abrir programas de Windows

```

java

import java.util.Scanner;

public class MenuProgramas {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=== MENÚ DE PROGRAMAS ===");
        System.out.println("1. Bloc de notas (notepad)");
        System.out.println("2. Calculadora (calc)");
        System.out.println("3. Paint (mspaint)");
        System.out.println("4. Explorador de archivos (explorer)");
        System.out.println("5. Salir");
        System.out.print("Elige una opción: ");

        int opcion = sc.nextInt();

        try {
            String programa = "";

            switch(opcion) {
                case 1:

```

```

        programa = "notepad.exe";
        break;
    case 2:
        programa = "calc.exe";
        break;
    case 3:
        programa = "mspaint.exe";
        break;
    case 4:
        programa = "explorer.exe";
        break;
    case 5:
        System.out.println("Saliendo...");
        return;
    default:
        System.out.println("Opción no válida");
        return;
}

System.out.println("Abriendo " + programa + "...");
ProcessBuilder pb = new ProcessBuilder(programa);
Process p = pb.start();

System.out.println("Programa iniciado. PID: " + p.pid());

} catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
}
}
}

```

✦ Ejercicio 4: Abrir página web en el navegador

```

java

import java.io.*;
import java.util.Scanner;

public class AbrirNavegador {
    public static void main(String[] args) throws IOException, InterruptedException {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

System.out.print("¿Qué página web quieres abrir? ");
System.out.println("Ejemplos: google.com, youtube.com, github.com");
String url = sc.nextLine();

// Asegurarse de que tenga http://
if (!url.startsWith("http://") && !url.startsWith("https://")) {
    url = "https://" + url;
}

System.out.println("Abriendo: " + url);

try {
    ProcessBuilder pb;

    // Diferentes comandos según el sistema operativo
    String os = System.getProperty("os.name").toLowerCase();

    if (os.contains("win")) {
        // Windows
        pb = new ProcessBuilder("cmd.exe", "/c", "start", url);
    } else if (os.contains("mac")) {
        // Mac
        pb = new ProcessBuilder("open", url);
    } else {
        // Linux
        pb = new ProcessBuilder("xdg-open", url);
    }

    Process p = pb.start();
    System.out.println("Navegador abierto");

} catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
    System.out.println("Intenta abrirlo manualmente: " + url);
}
}

```

 Ejercicio 5: Temporizador para abrir programas

```
java
```

```
import java.util.Scanner;
```

```
public class TemporizadorPrograma {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);

        System.out.println("=== TEMPORIZADOR DE PROGRAMA ===");
        System.out.print("¿Qué programa quieres abrir? (notepad/calc/mspaint): ");
        String programa = sc.nextLine() + ".exe";

        System.out.print("¿En cuántos segundos quieres que se abra? ");
        int segundos = sc.nextInt();

        System.out.println("El programa se abrirá en " + segundos + " segundos...");

        // Esperar los segundos indicados
        for (int i = segundos; i > 0; i--) {
            System.out.println("Tiempo restante: " + i + " segundos");
            Thread.sleep(1000); // Esperar 1 segundo
        }

        System.out.println("¡Abriendo " + programa + "!");

        // Abrir el programa
        ProcessBuilder pb = new ProcessBuilder(programa);
        Process p = pb.start();

        System.out.println("Programa abierto con éxito");
    }
}
```

Ejercicio 6: Abrir varios programas a la vez

```
java
```

```
public class MultiplesProgramas {
    public static void main(String[] args) {
        String[] programas = {"notepad.exe", "calc.exe", "mspaint.exe"};
```

```

        System.out.println("Abriendo " + programas.length + " programas...");

        try {
            for (int i = 0; i < programas.length; i++) {
                System.out.println((i+1) + ". Abriendo " + programas[i]);
                ProcessBuilder pb = new ProcessBuilder(programas[i]);
                pb.start();
                Thread.sleep(1000); // Esperar 1 segundo entre cada uno
            }

            System.out.println("¡Todos los programas abiertos!");
            System.out.println("Ciérralos manualmente cuando quieras");

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

Ejercicio 7: Crear y abrir archivo de texto automáticamente

```

java

import java.io.*;
import java.util.Scanner;

public class CrearYAbrir {
    public static void main(String[] args) throws IOException, InterruptedException {
        Scanner sc = new Scanner(System.in);

        System.out.print("Nombre del archivo (sin extensión): ");
        String nombre = sc.nextLine() + ".txt";

        System.out.print("Contenido del archivo: ");
        sc.nextLine(); // Limpiar buffer
        String contenido = sc.nextLine();

        // 1. Crear el archivo
        File archivo = new File(nombre);

        try (PrintWriter pw = new PrintWriter(archivo)) {

```

```

        pw.println("=== ARCHIVO CREADO DESDE JAVA ===");
        pw.println("Contenido: " + contenido);
        pw.println("Fecha: " + new java.util.Date());
        pw.println("=====");
    }

    System.out.println("Archivo creado: " + archivo.getAbsolutePath());

    // 2. Abrirlo con Notepad
    System.out.println("Abriendo con Bloc de notas...");
    ProcessBuilder pb = new ProcessBuilder("notepad.exe", archivo.getPath());
    Process p = pb.start();

    // 3. Esperar a que se cierre
    p.waitFor();
    System.out.println("Archivo cerrado");

    // Preguntar si borrar
    System.out.print("¿Borrar el archivo? (si/no): ");
    String respuesta = sc.nextLine();

    if (respuesta.equalsIgnoreCase("si")) {
        if (archivo.delete()) {
            System.out.println("Archivo borrado");
        } else {
            System.out.println("No se pudo borrar el archivo");
        }
    }
}
}

```