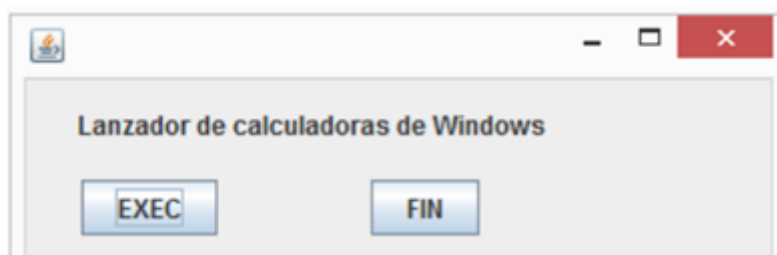


## EJERCICIO 1: Programa que lanza varios procesos

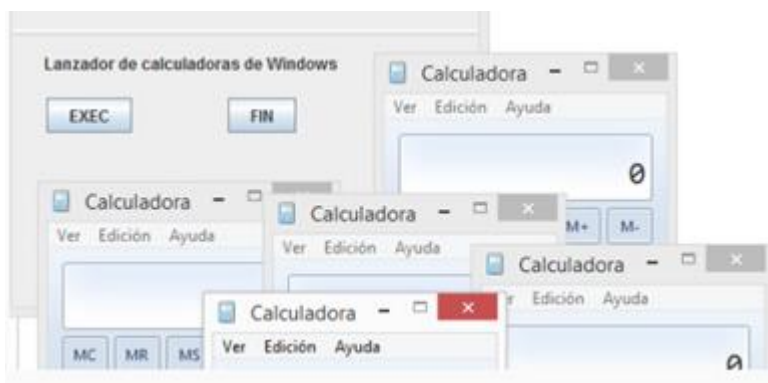
Aplicación Java que lanza varias calculadoras a la vez, lo que implica crear tantos procesos de la calculadora como queramos usar.

Pasos:

1. Crea un proyecto NetBeans que ejecuta la calculadora lanzando un proceso que ejecuta el comando Calc. Genera la aplicación "calculadora.jar".
2. Crea un proyecto Netbeans "lanzador" que permita ejecutar varios procesos de "calculadora.jar". Para ello, nuestra aplicación "lanzador" tendrá estas características:
  - Será una ventana con dos botones:
    - EXEC: que lanzará tres procesos de "calculadora.jar" cada vez que se pulse.
    - FIN: finalizará la aplicación



3. Genera la aplicación "lanzador.jar" y prueba su ejecución desde la línea de comandos. Haz una captura de pantalla de la ejecución del comando.



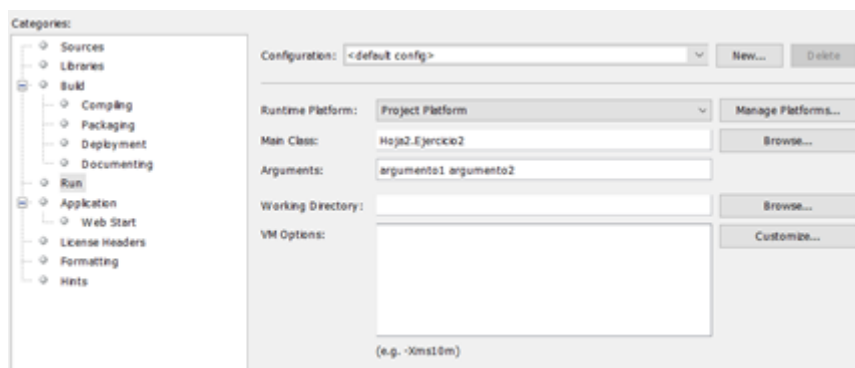
## EJERCICIO 2: El comando ping desde Java

Realiza un programa Java que ejecute en Windows el comando de MS-DOS "CMD /C PING *ip*" que nos dice si un recurso de la red está accesible o no. El programa tendrá que hacer ping a un rango de IPs consecutivas.

Tendrás que hacer los siguientes pasos:

1. El programa *MiPing* recibirá tres argumentos: la primera **ip** a hacer ping del rango, el **número de Ips** del rango y un nombre de **fichero** donde guardar la salida de los pings.

- Para pasar argumentos a un programa desde NetBeans deberás ir al menú: Run -> Set Project Configuration ->Customize...
- En el apartado run, en arguments deberás pasar los argumentos separados por un espacio.



Muy importante ejecutar el proyecto pinchando en el botón de Play verde de NetBeans para que coja los argumentos.



2. Deberás lanzar tantos comandos pings como IPs haya en el rango. Por ejemplo, si la ip comienza por la 142.250.201.67 y hay que hacer cinco pings, habrá que lanzar un proceso nuevo para los pings:

ping 142.250.201.67

ping 142.250.201.68

ping 142.250.201.69

ping 142.250.201.70

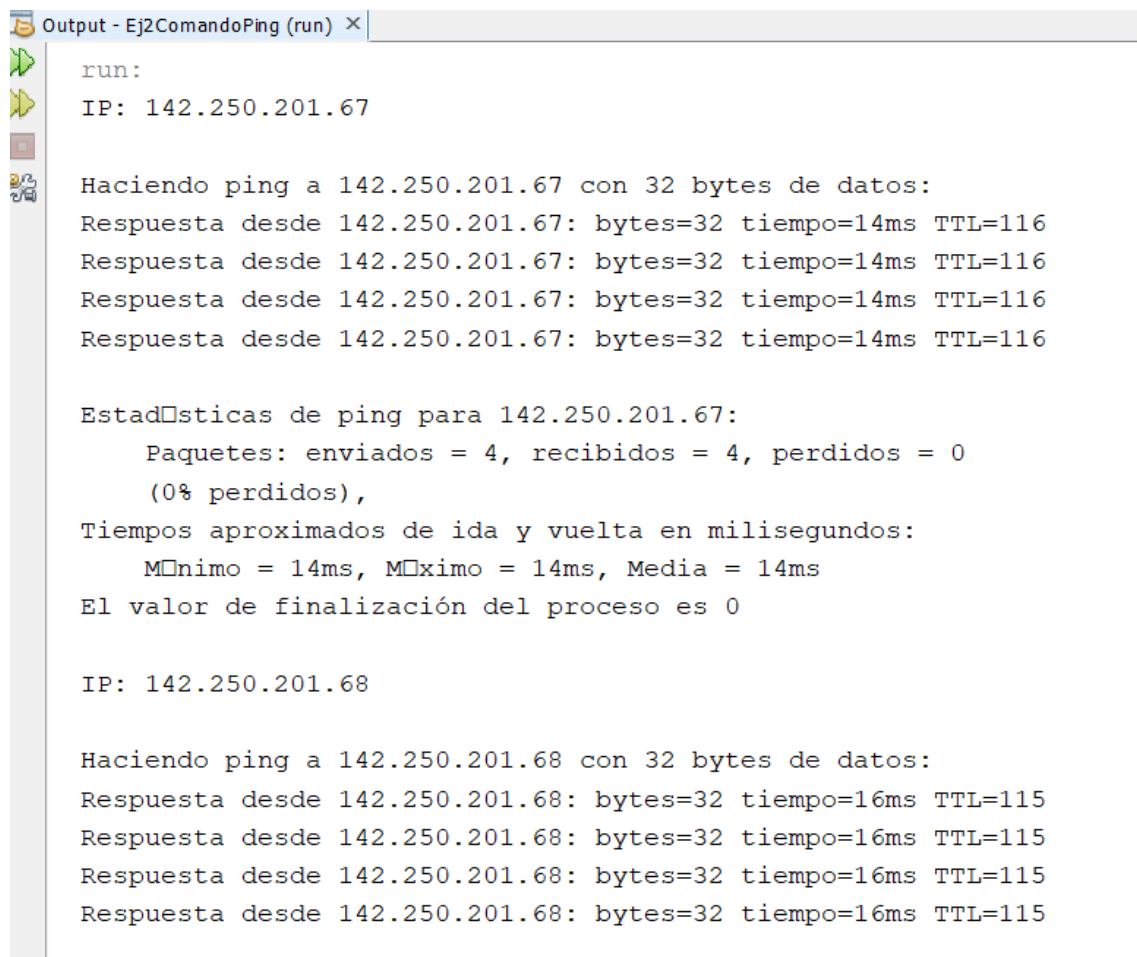
ping 142.250.201.71

3. Por cada proceso lanzado en el programa principal (main), se leerá lo que devuelve el proceso línea a línea (**readLine()** de la clase `BufferedReader`) para imprimirlo por pantalla y en un fichero de texto.

- Para obtener un flujo de entrada conectado con la salida del proceso hijo, deberás utilizar el método **getInputStream()** de la clase `Process`.

4. El proceso principal (main) deberá esperar a que termine la ejecución de cada proceso hijo y obtener el valor de finalización del proceso y mostrarlo por pantalla.

- Para que el proceso principal (main) espere a los procesos hijos tendrás que utilizar el método **waitFor()** de la clase `Process`.



```
run:
IP: 142.250.201.67

Haciendo ping a 142.250.201.67 con 32 bytes de datos:
Respuesta desde 142.250.201.67: bytes=32 tiempo=14ms TTL=116
Respuesta desde 142.250.201.67: bytes=32 tiempo=14ms TTL=116
Respuesta desde 142.250.201.67: bytes=32 tiempo=14ms TTL=116
Respuesta desde 142.250.201.67: bytes=32 tiempo=14ms TTL=116

Estadísticas de ping para 142.250.201.67:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 14ms, Máximo = 14ms, Media = 14ms
    El valor de finalización del proceso es 0

IP: 142.250.201.68

Haciendo ping a 142.250.201.68 con 32 bytes de datos:
Respuesta desde 142.250.201.68: bytes=32 tiempo=16ms TTL=115
Respuesta desde 142.250.201.68: bytes=32 tiempo=16ms TTL=115
Respuesta desde 142.250.201.68: bytes=32 tiempo=16ms TTL=115
Respuesta desde 142.250.201.68: bytes=32 tiempo=16ms TTL=115
```

.....

5. Por último, obtén también la salida de error de estándar del proceso. En el caso de que el comando de MS-DOS no exista, se deberá imprimir la salida de error por pantalla.

## **Criterios de calificación**

Ejercicio 1 (4 puntos):

- Correcta codificación del proyecto que lanza el comando "calc.exe" – 1,5 puntos.
- Correcta codificación de la aplicación "lanzador" que lanza varios procesos – 1 punto.
- Correcta ejecución de la aplicación "Lanzador.jar desde línea de comandos"– 0.5 puntos.
- Tratamiento de todos los posibles errores y excepciones – 0.5 puntos.
- Buena estructuración de código y documentación – 0,5 puntos

Ejercicio 2 (6 puntos):

- Paso de argumentos necesarios al programa – 0,5 puntos
- Lanzamiento de los comandos ping – 1
- Obtención del flujo e impresión de todas las líneas por consola – 1
- Guardado de todas las líneas en un fichero – 1
- El proceso padre espera al proceso hijo e impresión del mensaje – 1
- Obtención e impresión de la salida de error estándar – 0,5
- Tratamiento de todos los posibles errores y excepciones – 0,5 puntos
- Buena estructuración de código y documentación – 0,5 puntos

### **Indicaciones de entrega.**

Debes crear una carpeta para la tarea e incluye los proyectos y documentos correspondientes. Comprímelo todo en un fichero.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1\_apellido2\_nombre\_SIGxx\_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la primera unidad del MP de PSP, debería nombrar esta tarea como...

sanchez\_manas\_begona\_PSP01\_Tarea