

---

# Práctica 4: Interfaz de ventana de la TPMV

---

**Fecha de entrega:** 21 de Abril de 2013. 14:00h

**OBJETIVO:** Interfaces gráficas de usuario en Swing. Ventanas y componentes de Swing

## 1. Nueva funcionalidad

Podemos decir que con la práctica anterior ya tenemos una implementación de nuestra TPMV totalmente funcional. La única pega que le podemos poner es la facilidad de uso: en el modo interactivo, útil para ejecutar y depurar programas, la interfaz de consola no es demasiado amigable.

En esta práctica vamos a extender la aplicación de forma que exista un nuevo modo de ejecución adicional a los ya existentes `batch` e `interactive`. Es el modo `window`, en el que se presenta un interfaz de ventana que permite realizar las mismas operaciones que permitía el modo interactivo de consola.

A todos los efectos, el modo `window` es similar al modo `interactive` en el sentido de que el programa-mv se ejecutará bajo el control del usuario de la aplicación, según va interactuando con la aplicación (igual que en modo `interactive` el usuario va solicitando la ejecución de las siguientes instrucciones y/o cambiando el estado de la máquina).

## 2. Interfaz gráfica

El aspecto de la interfaz gráfica será algo parecido a la Figura 1.

En esta interfaz distinguimos los siguiente elementos:

- Una zona donde aparece el programa-mv cargado en la máquina. También hay una pista visual para saber cual es la siguiente instrucción a ejecutar.
- Estado de la pila de operandos y posibilidad de añadir y quitar elementos.



Figura 1: Apariencia de la interfaz gráfica

- Estado de la memoria de la máquina y posibilidad de alterar sus valores<sup>1</sup>.
- Caracteres que sirven de entrada al programa-mv cargado. Si al arrancar la aplicación no se especifica `-i`, no habrá nada en la entrada (igual que ocurre en el modo `interactive`). En caso de indicar fichero, en ese área de texto aparecerá su contenido completo para que el usuario pueda ver cuál será la entrada que el programa-mv procesará. A medida que éste va consumiendo caracteres (al ir ejecutando instrucciones `IN`), aparecerá una pista visual indicando que esos caracteres ya han sido leídos (la pista consiste en la sustitución de los caracteres leídos por el carácter `*`). En la figura 1 se puede ver la ejecución del programa que elimina la letra “a” de la entrada en el momento en el que ya ha leído la cadena “Esta es la ent” que ha sido sustituida por el `*`.
- Salida del programa: según se van ejecutando instrucciones `OUT` se van escribiendo ahí.
- Una zona superior a modo de “barra de herramientas” que contiene un botón para ejecutar la siguiente instrucción y otro botón para ejecutar el programa completo. Una vez que la ejecución del programa-mv ha terminado, la pulsación de esos dos botones no tendrá efecto. Existe también un botón para terminar la aplicación; cuando se pulsa se pide confirmación de la acción antes de terminar (con código de retorno 0).

<sup>1</sup>En la figura 1 se ve ese espacio vacío debido a que el programa-mv en ejecución no necesita almacenar ningún valor en ella.

Si durante la ejecución del programa-mv se produce algún error, aparecerá un cuadro de diálogo mostrando el mensaje correspondiente.

### 3. Entrega de la práctica

La práctica debe entregarse utilizando el mecanismo de entregas del campus virtual, no más tarde de la fecha indicada en la cabecera de la práctica.

Es indispensable que el código fuente supere los tests de unidad proporcionados y que el fichero enviado pase el programa validador publicado.

Sólo uno de los dos miembros del grupo debe hacerlo, subiendo al campus un fichero llamado `GrupoNN.zip`, donde NN representa el número de grupo con dos dígitos.

El fichero debe tener al menos el siguiente contenido<sup>2</sup>:

- Directorio `src` con el código de todas las clases de la práctica.
- Fichero `alumnos.txt` donde se indicará el nombre y apellidos de los componentes del grupo.

---

<sup>2</sup>Puedes incluir también opcionalmente los ficheros de información del proyecto de Eclipse