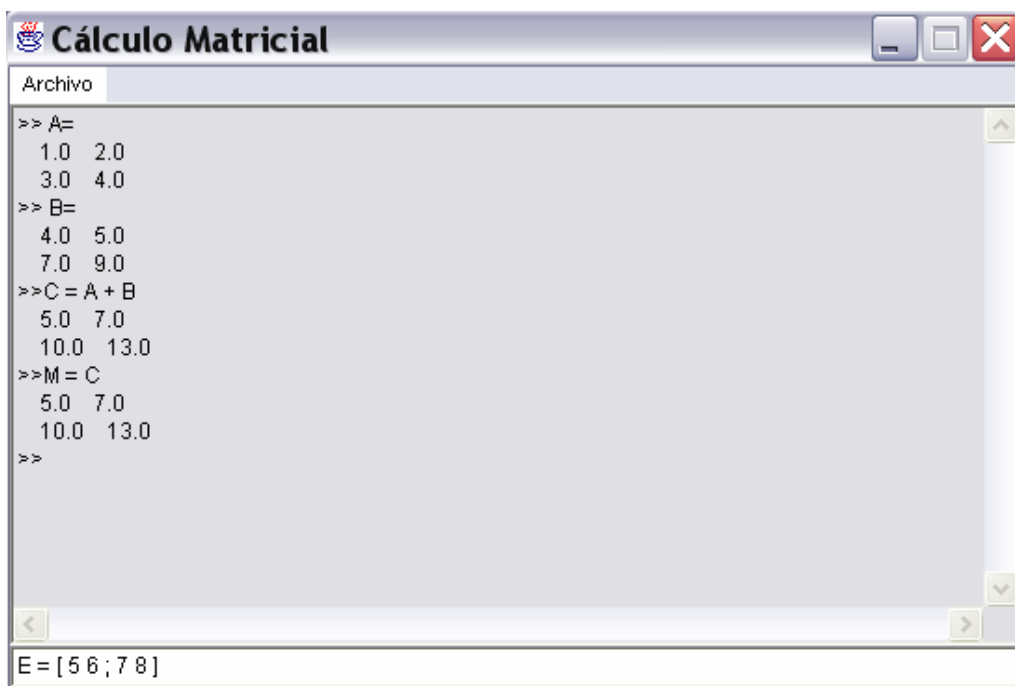


## EXAMEN DE SISTEMAS INFORMÁTICOS INDUSTRIALES

FEBRERO 2003

### BLOQUE I. PROGRAMACIÓN EN JAVA

Realizar un programa en Java que permita trabajar con matrices cuadradas de números reales y posea la apariencia que se muestra a continuación:



```
Archivo
>> A=
  1.0  2.0
  3.0  4.0
>> B=
  4.0  5.0
  7.0  9.0
>> C = A + B
  5.0  7.0
 10.0 13.0
>> M = C
  5.0  7.0
 10.0 13.0
>>
E = [ 5 6 ; 7 8 ]
```

Las características que debe poseer este programa son las siguientes:

- La ventana de la aplicación se divide en dos áreas. En la parte inferior se encuentra una zona editable en la que el usuario podrá introducir un comando relacionado con matrices. En el área superior, que no será editable, se mostrará el resultado de cada comando introducido en el área inferior cuando el usuario pulse *Intro*.
- La aplicación tendrá un menú denominado Archivo con las siguientes opciones:
  - Salir: Cerrará la aplicación
  - Limpiar pantalla: Limpiará el área superior de la aplicación
  - Borrar matrices: Borrará todas las matrices que se encuentran en memoria
- Las operaciones que podrán realizarse serán: definir matrices, visualizar matrices, copiar matrices, y sumar y restar matrices.



- ❑ Para definir matrices sólo se podrá utilizar nombre de matrices correspondientes a letras del alfabeto en mayúsculas. Además deberá existir un espacio entre los caracteres así como entre los números. Será necesario utilizar '[' y ']' para delimitar la matriz y ';' para separar las filas de la matriz. Por ejemplo, el usuario podría definir una matriz de 2x2 introduciendo el siguiente comando en el área inferior:

$$A = [ 1 2 ; 3 4 ]$$

No se deberá comprobar que la matriz introducida es cuadrada ni que la sintaxis es correcta. Cuando el usuario pulse *Intro* deberá mostrarse en el área superior la matriz introducida.

- ❑ Para visualizar matrices en el área superior, el usuario sólo deberá introducir el nombre de la matriz en el área inferior y pulsar *Intro*. Si se introduce el nombre de una matriz no definida deberá mostrarse en el área superior el mensaje "*Matriz no definida*".
- ❑ Para copiar matrices se utilizará el operador '=' dejando un espacio entre cada carácter. Al pulsar *Intro* se mostrará el resultado de la copia en el área superior. Si no existe la matriz de la que se quiere copiar deberá mostrarse en el área superior el mensaje "*Matriz no definida*".

Ejemplo: C = A.

- ❑ Al sumar o restar matrices se deberá comprobar que las dimensiones son correctas. Si las dimensiones no son correctas, deberá mostrarse en el área superior el mensaje "*Matrices incompatibles*". Deberá dejarse un espacio entre cada carácter. Al pulsar *Intro* se mostrará el resultado de la operación en el área superior. Si alguna matriz no está definida deberá mostrarse en el área superior el mensaje "*Matriz no definida*".

Ejemplo suma: D = A + B

Ejemplo resta: D = A – B

- ❑ La aplicación deberá contener como mínimo las siguientes clases:
  - Matriz.java: Definirá la clase **Matriz**, que permitirá definir matrices cuadradas de forma dinámica y realizar las operaciones que sean necesarias sobre matrices.
  - InterfazUsuario.java: Permitirá definir la interfaz de la aplicación.
  - CalculoMatricial.java: Contendrá el main de la aplicación.



A continuación se realiza la especificación detallada de la aplicación a implementar:

- Definición de la clase **Matriz** para trabajar con matrices cuadradas de números reales:
  - 1) Constructor por defecto (0.25 pts)
  - 2) Constructor con parámetro indicando la dimensión de la matriz (0.5 pts)
  - 3) Implementar todos los **métodos** necesarios de la clase:
    - i. *void introducirCadena(String mat)* (1.5 pts)

Almacenará en la matriz los elementos que se le pasen en forma de cadena según se ha indicado anteriormente.  
Por ejemplo: "A = [ 2 3 ; 4 5 ]"  
Deberá redimensionar la matriz si es necesario
    - ii. *String toCadena()* (1 pts)

Devolverá un *String* con la representación de la matriz. Por ejemplo:  
"2.0 3.0  
4.0 5.0"
    - iii. *void copia(Matriz mat)* (0.5 pts)

Crearé una matriz idéntica a la que se le pase como parámetro. Deberá redimensionar la matriz si es necesario
    - iv. *void suma(Matriz mat1, Matriz mat2)* (0.5 pts)

Crearé una matriz con la suma de las matrices que se le pasen  
Redimensionará la matriz si es necesario.
    - v. *void resta(Matriz mat1, Matriz mat2)* (0.5 pts)

Crearé una matriz con la resta de las matrices que se le pasen  
Redimensionará la matriz si es necesario.
- Definición de la clase **InterfazUsuario** para crear la interfaz de la aplicación y de las clases necesarias para gestionar los eventos que puedan producirse:
  - 4) Creación de la interfaz con las dos áreas especificadas junto con el menú Archivo. (0.5 pts)
  - 5) Gestionar los eventos del menú:
    - i. Salir (0.25 pts)
    - ii. Limpiar pantalla (0.25 pts)
    - iii. Borrar matrices (0.5 pts)





- 6) Gestionar el cierre de la aplicación (0.25 pts)
- 7) Visualizar matrices (0.75 pts)
- 8) Copiar matrices (0.75 pts)
- 9) Realizar operaciones matriciales (suma y resta) (1 pto)
- 10) Definir matrices (0.75 pts)
- Definición de la clase **CalculoMatricial** con el main para iniciar la aplicación (0.25 pts)

**Notas:**

- El proyecto debe guardarse en un directorio dentro de la unidad **c:** (Win95) o el directorio **c:\invitado** (Win NT)
- Se recomienda guardar de forma regular el contenido de los ficheros editados.
- Al finalizar el examen se entregará a cada alumno **un disco etiquetado** para almacenar el resultado. En la etiqueta el alumno escribirá su **nombre, apellidos y DNI**.
- En el disco **deberá** copiar todos los ficheros fuente (\*.java) desarrollados.

**Duración del examen: 3 horas**

