

# LABORATORIO DE PROGRAMACIÓN III (Curso 2012/2013)

## PRÁCTICA 1 FECHA LÍMITE: Viernes 8 de marzo

### Objetivos de la práctica:

1. Iniciación al entorno de Eclipse para programación en Java
2. Desarrollo de entornos gráficos de usuario usando Swing/Awt
3. Entrada/Salida en Java. Acceso a ficheros
4. Esquema divide y vencerás

### PARTE 1: CALCULADORA DE MATRICES

Diseña una aplicación Java que funcione como una calculadora de matrices. Las matrices se pueden introducir de dos formas distintas: (1) escribiendo los valores en la tabla que ofrece la aplicación (ver Figura 2) o (2) cargando sus valores desde fichero. La aplicación se utiliza a través de tres menús principales (ver Figuras):

- **Matrices:** Este menú permite configurar las matrices con las que vamos a operar a través de los siguientes submenús:
  - **Fijar tamaño:** Esta opción abre una nueva ventana que nos permite introducir el número de filas y de columnas de las matrices con las que vamos a operar (ver Figura 1). Se puede cambiar el tamaño en cualquier momento de la ejecución de la aplicación. Una vez fijado el tamaño, debes mostrar en la parte central del panel principal tres tablas de tipo `javax.swing.JTable` dentro de un panel tabulado de tipo `javax.swing.JTabbedPane`:
    - **Matriz 1:** En esta tabla se escribirán los valores de cada una de las posiciones de la matriz.  
1. Cuando la casilla está en blanco, se entiende que el valor es 0.
    - **Matriz 2:** En esta tabla se escribirán los valores de cada una de las posiciones de la matriz.  
2. Cuando la casilla está en blanco, se entiende que el valor es 0.
    - **Resultado:** En esta tabla se mostrará el resultado de aplicar una operación a las matrices 1 y 2.
  - **Cargar tablas:** Al seleccionar esta opción, procederemos a leer las dos tablas anteriores (que contienen las matrices 1 y 2) para realizar a continuación alguna operación con ellas. Si posteriormente las tablas se modifican y no se vuelve a seleccionar la opción “cargar tablas”, estas modificaciones no se tendrán en cuenta al efectuar las operaciones.
  - **Inicializar:** Borrado de las tablas e inicialización a cero de las dos matrices.
- **Operar:** Este menú nos permite efectuar 3 operaciones:
  - **Suma:** Suma las dos matrices y muestra el resultado en la tabla **Resultado**.
  - **Resta:** Lo mismo para la resta.
  - **Multiplicación:** Este submenú permite seleccionar la multiplicación normal o la basada en el algoritmo de Strassen (ver parte 2).
- **Fichero:** Este menú nos permite leer las matrices desde fichero y guardar las modificaciones a través de los submenús:
  - **Leer:** A su vez, muestra un submenú para leer la Matriz 1 y la 2. Al seleccionarlos, podemos navegar por el sistema de archivos para seleccionar el fichero deseado donde se encuentra almacenada la información.
  - **Guardar:** Lo mismo para guardar la información en archivo.
  - **Default:** El sistema da opción a cargar las matrices por defecto a partir de los ficheros existentes “matriz1.txt” y “matriz2.txt” en vez de tener que navegar por el sistema de archivos para seleccionarlos.

### Indicaciones sobre la interfaz gráfica de usuario

En primer lugar, diseñaremos la interfaz gráfica de usuario utilizando los componentes disponibles en `java.awt` y `javax.swing` como `JLabel`, `JButton`, `TextField`, etc. La aplicación principal será un marco de tipo `JFrame` con un menú asociado que muestre las diferentes opciones que se han explicado arriba.

En el menú **Fichero** utilizaremos el componente `javax.swing.JFileChooser` que nos permite navegar por el sistema de ficheros y seleccionar el fichero deseado.

La disposición de los componentes dentro de los paneles se realizará utilizando cualquiera de los gestores estándar (`GridLayout`, `BorderLayout`, etc.).

## Indicaciones sobre el acceso a ficheros

Para leer y escribir ficheros de texto, hay diferentes opciones en Java. Puedes utilizar las siguientes clases y sus métodos asociados:

- Para escribir desde un fichero, puedes utilizar `java.io.PrintWriter` que tiene una constructora que recibe como parámetro un `FileOutputStream` con el nombre del fichero a escribir.
- Para leer de fichero, puedes utilizar la clase `java.util.Scanner` que tiene una constructora que recibe como parámetro un `File` con el nombre del fichero a leer.

## Estructura de paquetes y de clases

Esta primera parte de la aplicación, se organizará en dos paquetes **interfaz** y **matrices**. En el paquete **interfaz** almacenaremos el código relacionado con la creación de la interfaz gráfica de usuario e implementaremos el método **main** que permite ejecutar la aplicación. En el paquete **matrices** implementaremos el tipo matriz y las operaciones asociadas.

## PARTE 2: MULTIPLICACION STRASSEN

En la segunda parte de la práctica, implementaremos el algoritmo de Strassen, llamado así por Volker Strassen, que es un algoritmo usado para la multiplicación de matrices grandes. Es asintóticamente más rápido que el algoritmo de multiplicación de matrices estándar, pero para matrices pequeñas más lento que el algoritmo más rápido conocido. Puedes encontrar la descripción del algoritmo a implementar en: [http://es.wikipedia.org/wiki/Algoritmo\\_de\\_Strassen](http://es.wikipedia.org/wiki/Algoritmo_de_Strassen).

Los detalles de implementación se ampliarán en clase.

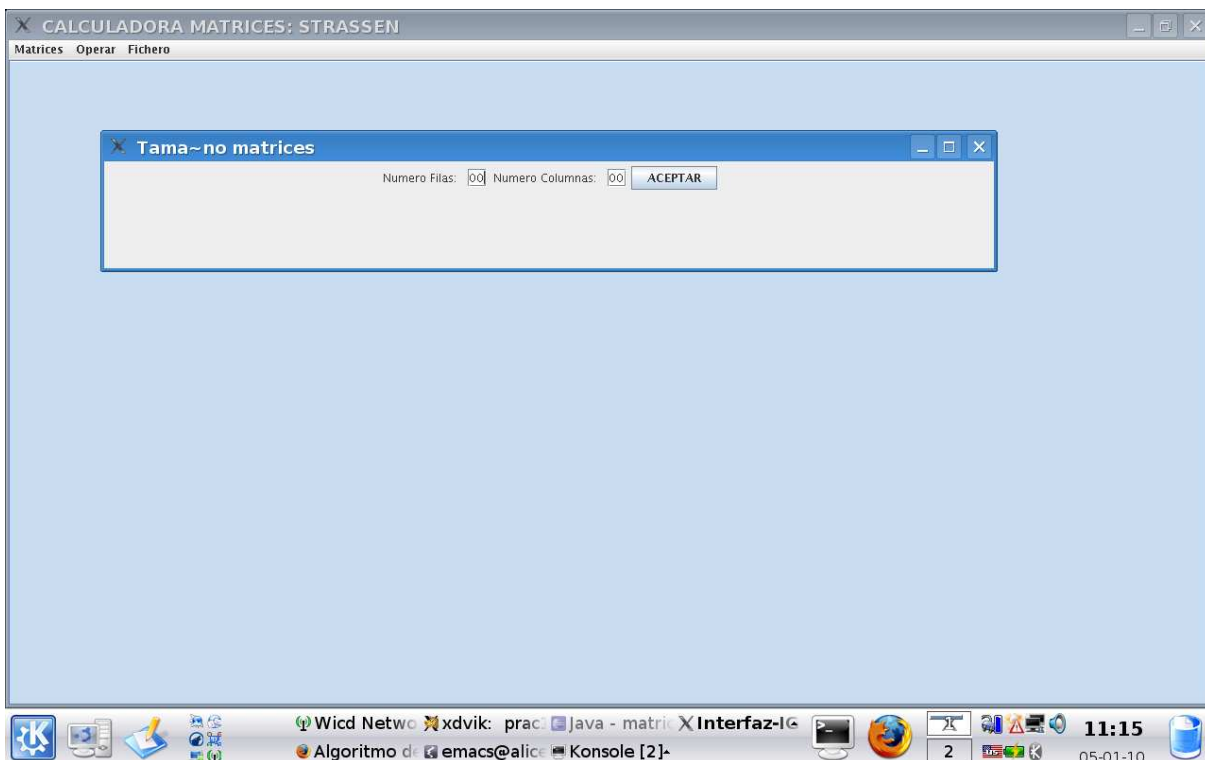


Figura 1: Frame para introducir tamaño de matrices

CALCULADORA MATRICES: STRASSEN											
Matrices Operar Fichero											
MATRIZ 1	MATRIZ 2	RESULTADO									
2	4	6	8	10	12	14	16	18	20	22	24
26	28	30	32	34	36	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figura 2: Vista de la aplicación con tablas