

<b>EVALUACIÓN</b>	Obligatorio 1	<b>GRUPO</b>	TODOS	<b>FECHA</b>	19 Abril 2022 V05
<b>MATERIA</b>	Programación II				
<b>CARRERA</b>	Ing. en Sistemas. Eléctrica, Electrónica, Telecomunicaciones, Lic. en Sistemas				
<b>CONDICIONES</b>	<p>- Puntos: Máximo: 20 Mínimo: 0</p> <p>- Lectura: <b>19 de abril de 2022</b></p> <p>- Fecha máxima de entrega: <b>17 de mayo de 2022</b></p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP o RAR</p> <p><b>IMPORTANTE:</b></p> <p>- Inscribirse.</p> <p>- Formar grupos de hasta dos personas del <b>mismo</b> dictado.</p> <p>- Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO".</p> <p>- Verificar que el trabajo quedó correctamente subido.</p>				

# MATRIZ

## PARTE 1)

Se desea implementar en Java la clase Matriz que tiene como atributo una matriz de trabajo de enteros y ofrezca diferentes métodos sobre dicha matriz numérica:

1	public void cargarMatrizEsquina(int n, int desde)	<p>Crea una matriz cuadrada de <math>n \times n</math>, cargada en forma de "esquina" con el valor "desde" en la posición 0, 0 y la almacena como matriz de trabajo. Ver ejemplo.</p> <p>Ejemplo de matriz esquina con <math>n=5</math> y <math>desde=10</math></p> <pre> 10 11 12 13 14 11 11 12 13 14 12 12 12 13 14 13 13 13 13 14 14 14 14 14 14 </pre>
2	public void cargarPuntas(int n)	<p>Crea una matriz cuadrada de <math>n \times n</math> (<math>n</math> par), cargada según el patrón y la almacena como matriz de trabajo. Ver ejemplo.</p> <p>Ejemplo: <math>n = 8</math></p> <pre> 4 3 2 1 1 2 3 4 3 2 1 0 0 1 2 3 2 1 0 0 0 0 1 2 1 0 0 0 0 0 1 1 0 0 0 0 0 1 2 1 0 0 0 1 2 3 2 1 0 0 1 2 3 4 3 2 1 1 2 3 4 </pre>

3	public boolean esConectada()	Debe retornar true si todos los números 0 de la matriz de trabajo son adyacentes en forma vertical u horizontal. Si no hay 0 o no cumple, no es conectada. Ejemplo de matriz conectada: 1 0 1 4 5 6 0 0 0 0 3 3 4 0 0 3 4 -2 0 0 0 0 0 5 0 6 7 0 0 0 0 0
4	public void generarMolino(int n)	Crea una matriz cuadrada de $n \times n$ ( $n$ par), cargada según “aspas” de un molino y la asigna como matriz de trabajo. Ver ejemplo. Ejemplo $n=6$ : 1 1 1 0 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 0 1 0 0 1 1 1
5	public int[][] getMatriz()	Retorna la matriz de trabajo.
6	public Matriz()	El constructor por defecto crea una matriz de $4 \times 5$ , iniciada en números randómicos entre 0 y 10 (ambos inclusive).
7	public String menorComunFilas()	Retorna un string que contiene el menor elemento común a todas las filas de la matriz de trabajo. Si no hay, retorna el string "NO TIENE". Ejemplo: 1 5 8 4 2 5 2 0 7 9 -3 8 5 -2 2 Retorna "2"
8	public boolean reacomodar(int m, int n)	Crea una nueva matriz, que es el resultado de reacomodar recorriendo por filas los elementos en la cantidad $m$ de filas y $n$ de columnas y la asigna como matriz de trabajo. Si no se puede reacomodar, no se modifica la matriz de trabajo. Retorna true si se pudo modificar. Ejemplo: $m=1$ $n=4$ , siendo la matriz de trabajo: 1 2 3 4 Se obtiene como matriz de trabajo: [1 2 3 4]. Retorna true.
9	public void setMatriz(int[][] mat)	Recibe una matriz y la almacena como matriz de trabajo.
10	public boolean tieneFilaIgual()	Retorna verdadero si hay alguna fila de la matriz de trabajo que tenga todos sus números iguales. Ejemplo: 4 2 5 1 2 2 2 2 2 2 5 2 -1 6 7 Retorna true.

Nota: de ser necesario, se pueden crear otros métodos privados. **No se puede modificar la firma de los métodos pedidos.**

## PARTE 2)

Implementar en Java en la clase Prueba el main para ejemplificar el uso de la clase Matriz ofreciendo un menú de consola que permita usar las veces deseadas las opciones que se detallan más abajo.

IMPORTANTE: el menú **DEBE** utilizar las letras indicadas para cada opción, y aceptar minúsculas o mayúsculas.

El menú a presentar es similar al siguiente:

```
Ingrese la opción
A: cambiar entrada      B: cambiar salida      C: cargar datos      D: mostrar reborde      E: mostrar tabulación      F: esquina
G: puntas              H: ver conectada      I: molino           J: ver menor común      K: reacomodar          L: ver iguales
X: terminar
```

El programa crea un objeto de la clase Matriz (usando el método 6) y sobre ese objeto se trabajará desde el menú. Inicialmente la entrada y salida es por consola.

Detalle de las opciones:

- cambiar la forma de entrada.** Al ingresar a esta opción, se modificará la forma de entrada: de ingresar por consola a ingresar desde el archivo “entrada.txt” (ubicado en la carpeta test del proyecto), o viceversa. (Ver nota al final).
- cambiar la forma de salida.** Al ingresar a esta opción, se modificará la forma de salida: de consola al archivo “salida.txt” (ubicado en la carpeta test del proyecto), o viceversa. (Ver nota al final).
- cargar matriz numérica para trabajo.** Se ingresa la cantidad de filas, cantidad de columnas y todos los elementos de la matriz por filas. Esta matriz se utilizará en las consultas. Usa el método 9-setMatriz.
- mostrar matriz de trabajo con reborde.** Utiliza el método 5-getMatriz para obtener la matriz y la muestra. Ejemplo: (observar alineación y formato del reborde)

```
+-----+-----+-----+-----+-----+-----+-----+
|      1|      1232|      3|      4|      5|      6|      7|      8|
+-----+-----+-----+-----+-----+-----+-----+
|      8|      7|      6|     -45|      4|      3|      2|      1|
+-----+-----+-----+-----+-----+-----+-----+
|      0|      2|      0|      4|     150|      5|      0|      6|
+-----+-----+-----+-----+-----+-----+-----+
|      3|      3|      4|     114|      5|     1515|      6|      6|
+-----+-----+-----+-----+-----+-----+-----+
|    -45661|      2|      621|      4|      1|      5|      1|      6|
+-----+-----+-----+-----+-----+-----+-----+
```

- mostrar matriz de trabajo con tabulación.** Utiliza el método 5-getMatriz para obtener los datos y la muestra. Ejemplo: (observar alineación a la derecha).

```
      1      1232      3      4      5      6      7      8
      8      7      6     -45      4      3      2      1
      0      2      0      4     150      5      0      6
      3      3      4     114      5     1515      6      6
    -45661      2      621      4      1      5      1      6
```

- cargar Matriz Esquina.** Se ingresa el valor de n y desde qué valor iniciar y se aplica en la matriz de trabajo el método 1-cargarMatrizEsquina.
- cargar Puntas.** Se ingresa el valor n (debe ser par) y se aplica en la matriz de trabajo el método 2-cargarPuntas.
- es Conectada.** Se muestra si la matriz de trabajo es conectada o no, usando el método 3- esConectada.
- generar Molino.** Se ingresa el valor de n (debe ser par) y se aplica en la matriz de trabajo el método 4-generarMolino.
- menor Comun Filas.** Muestra el menor elemento común de la matriz de trabajo, utiliza el método 7-menorComunFilas
- reacomodar.** Se ingresa la cantidad de filas y columnas. Si es posible, se reacomoda la matriz de trabajo e informa si se pudo realizar, utiliza el método 8- reacomodar.
- tiene Fila Igual.** Muestra si la matriz de trabajo tiene o no fila igual, usando el método 10-tieneFilaIgual.
- termina el programa.**

## Notas

**Entrada:** Definir la variable: Scanner input;

- a) para tomar datos de consola realizar esta asignación:  
`input = new Scanner(System.in);`
- b) para tomar del archivo test\datos.txt, realizar esta asignación:  
`input = new Scanner(new File(".\\test\\datos.txt"));`

Se lee luego usando los métodos comunes de Scanner.

**Salida:** Definir las siguientes variables:

```
PrintStream consola = System.out;
PrintStream archivo = new PrintStream(new FileOutputStream(".\\test\\salida.txt"));
```

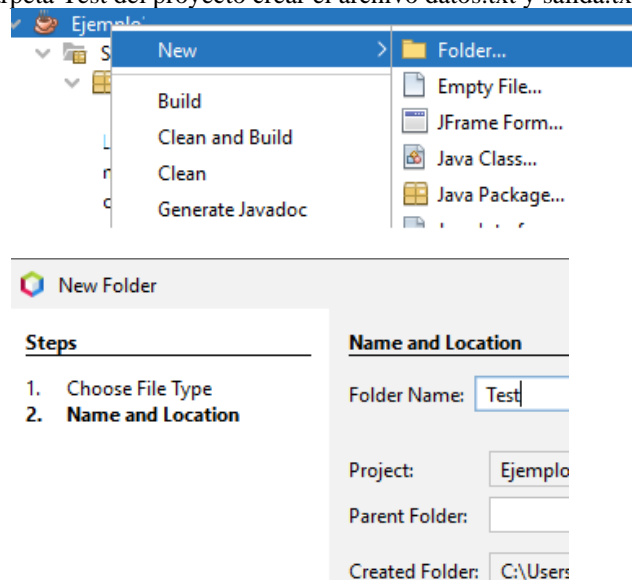
- a) para utilizar la consola, asignar:  
`System.setOut(consola);`
- b) para utilizar el archivo, asignar:  
`System.setOut(archivo);`

Se muestra luego usando los métodos comunes de System.out.println() y System.out.print().

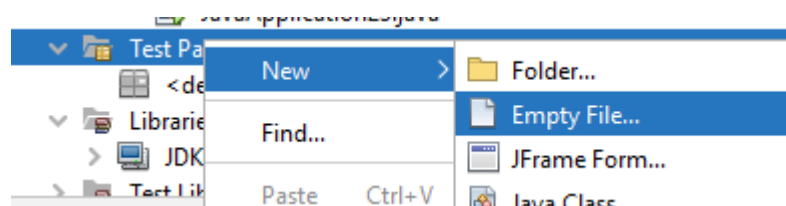
Tener presente que puede ser necesario alternar entre pestañas para que actualice el archivo de salida.txt.

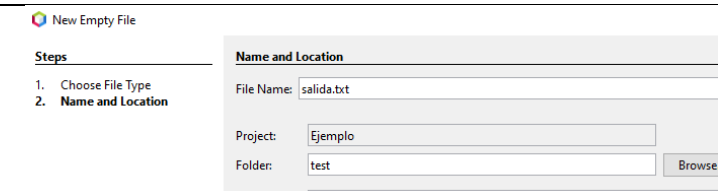
En todos los casos, agregar los imports y manejo de errores/excepciones correspondiente.

En la carpeta Test del proyecto crear el archivo datos.txt y salida.txt. Si no se dispone la carpeta Test, crearla:



Luego crear los dos archivos en esa carpeta. Ejemplo de creación de salida.txt:





En forma similar, agregar el otro archivo:

Según la versión de NetBeans puede haber pequeñas diferencias en la interfaz presentada.

### Entrega:

La entrega consiste de un archivo zip de hasta 40MB que contenga:

a) **carpeta del código.** Dentro debe estar el proyecto completo en NetBeans, incluyendo todos los fuentes Java. **IMPORTANTE:** En el código fuente de cada clase debe estar el nombre de los autores. Se tendrá especial consideración acerca de la calidad del código (reusabilidad, lógica, estilo de codificación, uso de Java, etc.)

b) **UNICO pdf** que contenga:

- 1) Carátula con foto, nombre y número de estudiante de los 2 integrantes del equipo. Las 2 personas deben pertenecer al mismo grupo de clase.
- 2) Representación en UML de la clase Matriz, con todos los métodos y atributos.

### RECORDATORIO: IMPORTANTE PARA LA ENTREGA

#### ➤ **Obligatorios** (Cap.IV.1, Doc. 220)

La entrega de los obligatorios será en formato digital online. Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. La entrega se realizará desde [gestion.ort.edu.uy](http://gestion.ort.edu.uy)
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. **Sugerimos realizarlo con anticipación.**
3. Cada equipo debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
4. El archivo a subir debe tener **un tamaño máximo de 40mb.**
5. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el '**grupo de obligatorio**'.
6. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
7. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, etc)
8. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta **antes de las 20:00hs** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.