

# UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA MINAS, GEOLOGÍA Y CIVIL

ESCUELA DE FORMACION PROFESIONAL DE INGENIERÍA DE SISTEMAS



## INFORME DE LABORATORIO 04

### MATRICES

09 octubre 2019

Jueves 2 - 5 pm

ASIGNATURA: IS142 Programación Orientada a Objetos

DOCENTE: M.sC. Ing. Fredy Barrientos

ALUMNO: Ramos Conga, Christian Yordi

AYACUCHO-PERÚ  
2019

## LABORATORIO 005

### 1. Objetivos

#### 1.1. Objetivo General

Poner en práctica el concepto de Matrices en Java.

#### 1.2. Objetivos Específicos

☑ Declarar, instanciar, inicializar, asignar y leer datos de una matriz.

☑ Iterar por medio de un ciclo for anidado los elementos de una matriz.

### 2. Requisitos

Este laboratorio requiere de conocimientos previos de manejo de variables, cadenas, operadores, estructuras de control y arreglos de una dimensión en Java.

### 3. Conceptos relacionados

#### 3.1. Matrices en Java

Otra estructura de datos en Java que podemos utilizar es una matriz. A diferencia de un arreglo de una dimensión, una matriz podríamos verla como dos arreglos, un arreglo maneja los renglones y otro arreglo las columnas, y al juntarlos obtenemos una matriz. Sin embargo, veremos que existen muchas cosas similares con un arreglo, y una vez entendido el concepto de arreglo es más sencillo entender el concepto de matriz.

Por ejemplo, podemos definir una matriz de 4 renglones por 7 columnas, de tipo enteros, sin embargo, puede ser de cualquier tipo que definamos.

Podemos recuperar el largo de los renglones con el código `nombreArreglo.length` y podemos obtener el largo de las columnas escribiendo `nombreArreglo[0].length`, es decir, que con cualquier renglón válido seleccionado podemos obtener el largo de las columnas. Esto nos va a servir posteriormente para iterar por medio de un ciclo for anidado cada uno de los elementos de la matriz.

De igual manera que un arreglo, no todos los elementos de una matriz deben contener valores. Los valores que no tengan un valor asignado tendrán el valor por default según el tipo de datos definido para la matriz.

#### 3.2. Declaración de una matriz

A continuación, veamos la sintaxis para declarar una matriz:

tipo `[][]` nombreArreglo ó tipo `nombreArreglo [][]`;

### **3.3. Instanciar matrices**

A continuación, veremos cómo inicializar una matriz, ya que hasta el momento con sólo declarar una variable de tipo matriz la JVM no sabe cuan largo y ancho es la matriz, para ello debemos inicializarlo, veamos como:

```
nombreArreglo = new tipo[renglones][columnas];
```

### **3.4. Inicializar los elementos de una matriz**

A continuación, veremos cómo inicializar los elementos de una matriz:

```
nombreArreglo[índice_renglon][índice_columna] = valor;
```

### **3.5. Extraer elementos de una matriz**

A continuación, veremos cómo extraer elementos de una matriz:

```
variableReceptora = nombreArreglo[índice_renglon][índice_columna];
```

## **4. Desarrollo del laboratorio**

### **4.1. Matriz en Java**

Vamos a crear el proyecto de nombre Matrices. Los pasos para crear un proyecto ya lo conocemos:

## **5. Assignment**

**La Tarea Académica N° 04 (TA04) consiste en crear un proyecto de nombre TA04\_Nombre\_ApellidoPaterno, y el paquete debe llamarse matrices. A continuación, agregar una clase main para cada ejercicio, por ejemplo, para el ejercicio 1, la clase main debe llamarse Ejercicio001.**

La TA04 está compuesta por los siguientes ejercicios:

1. Realiza un programa que rellene un array de 6 filas por 10 columnas con números enteros positivos comprendidos entre 0 y 1000 (ambos incluidos). A continuación, el programa deberá dar la posición tanto del máximo como del mínimo.

```

6 package matrices;
7
8 /**
9  *
10  * @author Añapa
11  */
12 public class Ejercicio001 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18
19         int[][] num = new int[6][10];
20
21         int fila;
22         int columna;
23
24         int minimo = Integer.MAX_VALUE;
25         int filaMinimo = 0;
26         int columnaMinimo = 0;
27
28         int maximo = Integer.MIN_VALUE;
29         int filaMaximo = 0;
30         int columnaMaximo = 0;
31
32         System.out.print("\n      ");
33         for(columna = 0; columna < 10; columna++) {
34             System.out.print("    " + columna + "  ");
35         }
36         System.out.println();
37
38         System.out.print("      ⌈");
39         for(columna = 0; columna < 10; columna++) {
40             System.out.print("———");
41         }
42         System.out.println("⌋ ");
43
44         for(fila = 0; fila < 6; fila++) {
45             System.out.print("    " + fila + " | ");
46             for(columna = 0; columna < 10; columna++) {
47                 num[fila][columna] = (int)(Math.random() * 1001);
48                 System.out.printf("%5d ", num[fila][columna]);
49
50                 // Calcula el mínimo y guarda sus coordenadas
51                 if (num[fila][columna] < minimo) {
52                     minimo = num[fila][columna];
53                     filaMinimo = fila;
54                     columnaMinimo = columna;
55                 }
56             }
57             System.out.println();
58         }
59
60         System.out.println("Mínimo: " + minimo + " en fila " + filaMinimo + " columna " + columnaMinimo);
61         System.out.println("Máximo: " + maximo + " en fila " + filaMaximo + " columna " + columnaMaximo);
62     }
63 }

```

```

52         if (num[filas][columna] < minimo) {
53             minimo = num[filas][columna];
54             filasMinimo = filas;
55             columnaMinimo = columna;
56         }
57
58         // Calcula el máximo y guarda sus coordenadas
59         if (num[filas][columna] > maximo) {
60             maximo = num[filas][columna];
61             filasMaximo = filas;
62             columnaMaximo = columna;
63         }
64     }
65     System.out.println("  ");
66 }
67 System.out.print("    ");
68 for(columna = 0; columna < 10; columna++) {
69     System.out.print("-----");
70 }
71
72 System.out.println("\n\nEl máximo es " + maximo + " y está en la fila " + filasMaximo + ", columna " + columnaMaximo);
73 System.out.println("El mínimo es " + minimo + " y está en la fila " + filasMinimo + ", columna " + columnaMinimo);
74 }
75 // TODO code application logic here

```

Output

matrices (run) × matrices (run) #2 ×

run:

0 1 2 3 4 5 6 7 8 9

0	166	839	736	742	406	184	322	229	4	239
1	126	385	920	303	325	452	124	399	113	907
2	834	776	364	776	908	412	813	234	811	751
3	411	964	176	74	51	921	644	286	763	940
4	128	394	823	108	100	638	983	921	774	44
5	105	42	301	164	138	924	251	901	217	590

El máximo es 983 y está en la fila 4, columna 6  
El mínimo es 4 y está en la fila 0, columna 8  
BUILD SUCCESSFUL (total time: 0 seconds)

2. Escribe un programa que pida 20 números enteros. Estos números se deben introducir en un array de 4 filas por 5 columnas. El programa mostrará las sumas parciales de filas y columnas igual que si de una hoja de cálculo se tratara. La suma total debe aparecer en la esquina inferior derecha.

```

6 package matrices;
7 import java.util.Scanner;
8 /**
9  *
10  * @author Añapa
11  */
12 public class Ejercicio002 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18
19         int[][] num = new int[4][5]; // array de 4 filas por 5 columnas
20
21         int fila;
22         int columna;
23
24         // Lee los datos de teclado
25         System.out.println("\nPor favor, introduzca los números (enteros) en el array\n");
26         for(fila = 0; fila < 4; fila++) {
27             for(columna = 0; columna < 5; columna++) {
28
29
30
31
32                 num[fila][columna] = Integer.SIZE;
33             }
34         }
35
36         // Muestra los datos y las sumas parciales de las filas
37         int sumaFila;
38         for(fila = 0; fila < 4; fila++) {
39             sumaFila = 0;
40             for(columna = 0; columna < 5; columna++) {
41                 System.out.printf("%7d ", num[fila][columna]);
42                 sumaFila += num[fila][columna];
43             }
44             System.out.printf("|%7d\n", sumaFila);
45         }
46
47         // Muestra las sumas parciales de las columnas
48         for(columna = 0; columna < 5; columna++) {
49             System.out.print("-----");
50         }
51         System.out.println("-----");
52
53         int sumaColumna;
54         int sumaTotal = 0;

```

```

53     int sumaColumna;
54     int sumaTotal = 0;
55     for(columna = 0; columna < 5; columna++) {
56         sumaColumna = 0;
57         for(fila = 0; fila < 4; fila++) {
58             sumaColumna += num[fila][columna];
59         }
60
61         sumaTotal += sumaColumna;
62         System.out.printf("%7d  ", sumaColumna);
63     }
64     System.out.printf("|%7d  ", sumaTotal);
65 }
66 // TODO code application logic here
67

```

Output

matrices (run) × matrices (run) #2 ×

run:

Por favor, introduzca los números (enteros) en el array

32	32	32	32	32		160
32	32	32	32	32		160
32	32	32	32	32		160
32	32	32	32	32		160

---

128	128	128	128	128		640	BUILD SUCCESSFUL (total time: 0 seconds)
-----	-----	-----	-----	-----	--	-----	--

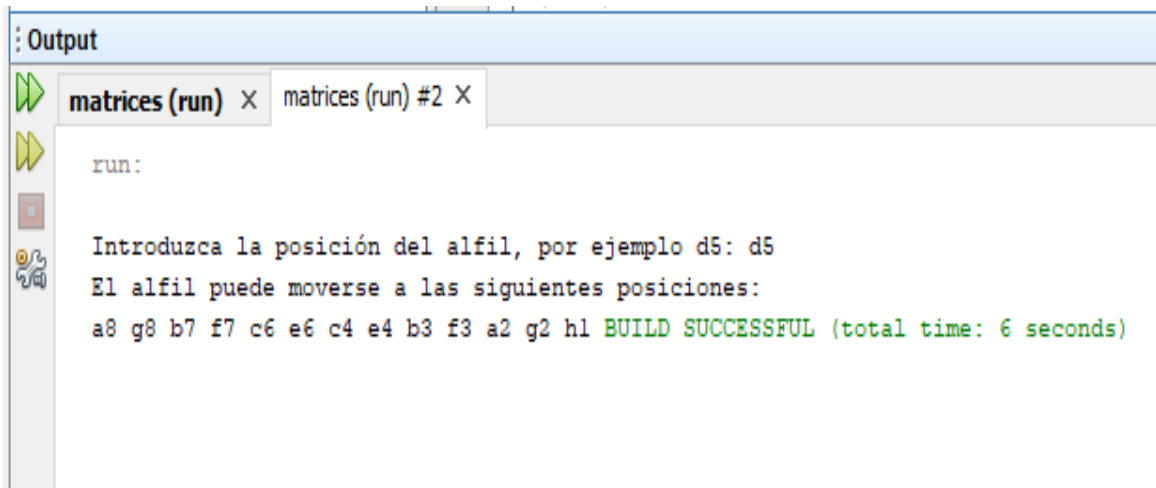
3. Escribe un programa que, dada una posición en un tablero de ajedrez, nos diga a qué casillas podría saltar un alfil que se encuentra en esa posición. Como se indica en la figura, el alfil se mueve siempre en diagonal. El tablero cuenta con 64 casillas. Las columnas se indican con las letras de la “a” a la “h” y las filas se indican del 1 al 8.

```

6 package matrices;
7 import java.util.Scanner;
8 /**
9  *
10  * @author Añapa
11  */
12 public class Ejercicio003 {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18
19         Scanner leer = new Scanner(System.in);
20
21
22
23         System.out.print("\nIntroduzca la posición del alfil, por ejemplo d5: ");
24
25         String posicionAlfil = leer.next();
26
27
28
29         int columnaAlfil = (int)(posicionAlfil.charAt(0)) - 96;
30         int filaAlfil = (int)(posicionAlfil.charAt(1)) - 48;
31
32         System.out.println("El alfil puede moverse a las siguientes posiciones:");
33
34         for(int fila = 8; fila >= 1; fila--) {
35             for(int columna = 1; columna <= 8; columna++) {
36                 if ((Math.abs(filaAlfil - fila) == Math.abs(columnaAlfil - columna))
37                     && (! ((filaAlfil == fila) && (columnaAlfil == columna)))) {
38                     System.out.print((char)(columna + 96) + " " + fila + " ");
39                 }

```





```
run:
Introduzca la posición del alfil, por ejemplo d5: d5
El alfil puede moverse a las siguientes posiciones:
a8 g8 b7 f7 c6 e6 c4 e4 b3 f3 a2 g2 h1 BUILD SUCCESSFUL (total time: 6 seconds)
```

Finalmente, debe subir el proyecto y el informe (.pdf) al repositorio remoto (Classroom de GitHub).

Url para activar el repositorio remoto: <https://classroom.github.com/a/dtddnILf>