



Módulo 1: Programación en Java

Roger F. González Camacho

Agosto 2023

1.4 Tipos de datos estructurados

- ❑ Arrays.
- ❑ Listas.
- ❑ Colecciones.

Arrays multidimensionales.

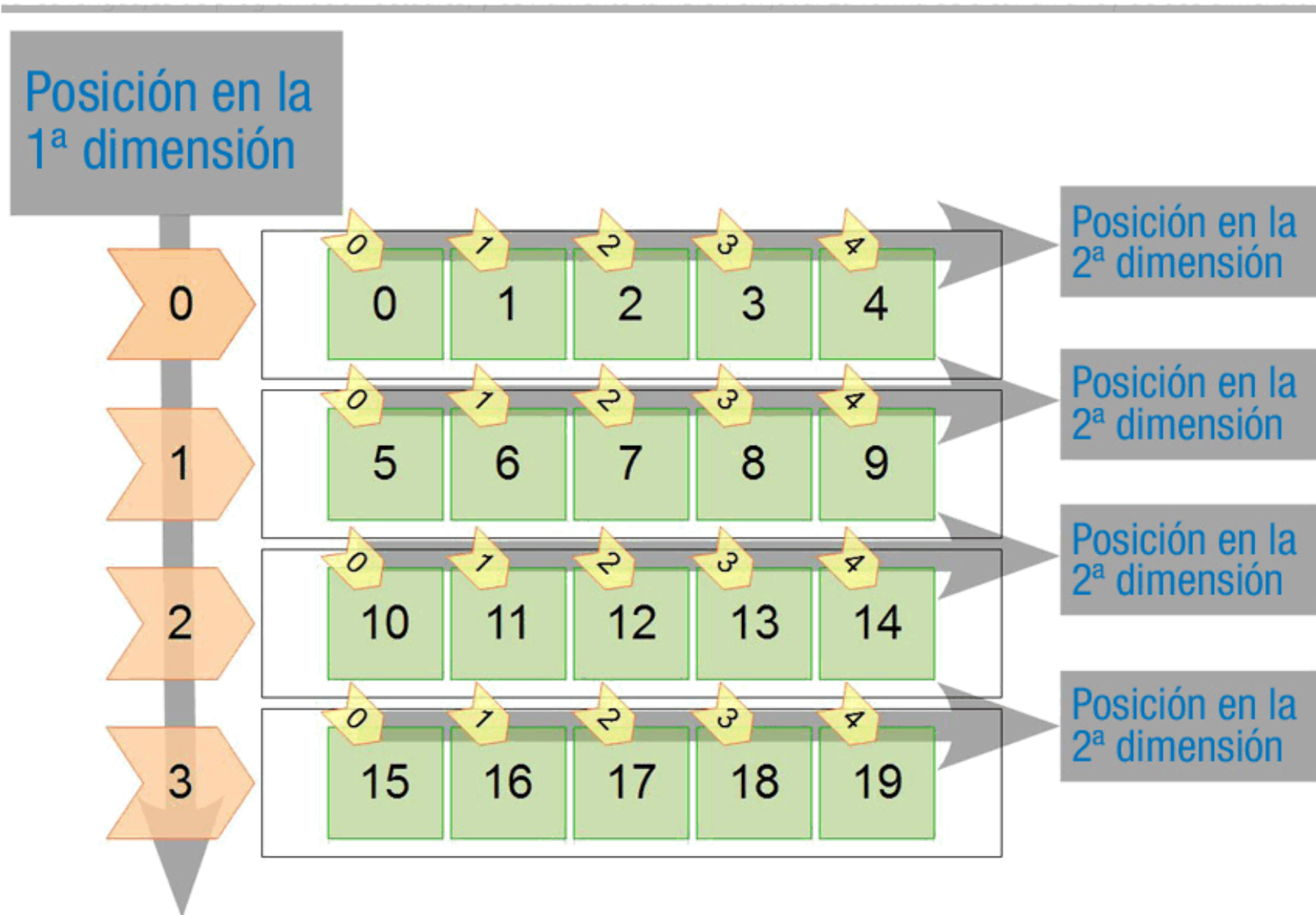
¿Qué estructura de datos utilizarías para almacenar los píxeles de una imagen digital?

Normalmente las imágenes son cuadradas así que una de las estructuras más adecuadas es la matriz. En la matriz cada valor podría ser el color de cada píxel. Pero, **¿qué es una matriz a nivel de programación?** Pues es un array con dos dimensiones, o lo que es lo mismo, un array cuyos elementos son arrays de números.

Los arrays multidimensionales están en todos los lenguajes de programación actuales, y obviamente también en Java. La forma de crear un array de dos dimensiones en Java es la siguiente:

```
int[ ][ ] a2d = new int[4][5];
```

Arrays multidimensionales.



Arrays multidimensionales.

```
int [][][] arrayde3dim;
```

```
Arrayde3dim = new int[2][3][4];
```

Uso.

```
int[ ][ ] a2d = new int[4][5];
```

```
a2d[0][0] = 3;
```

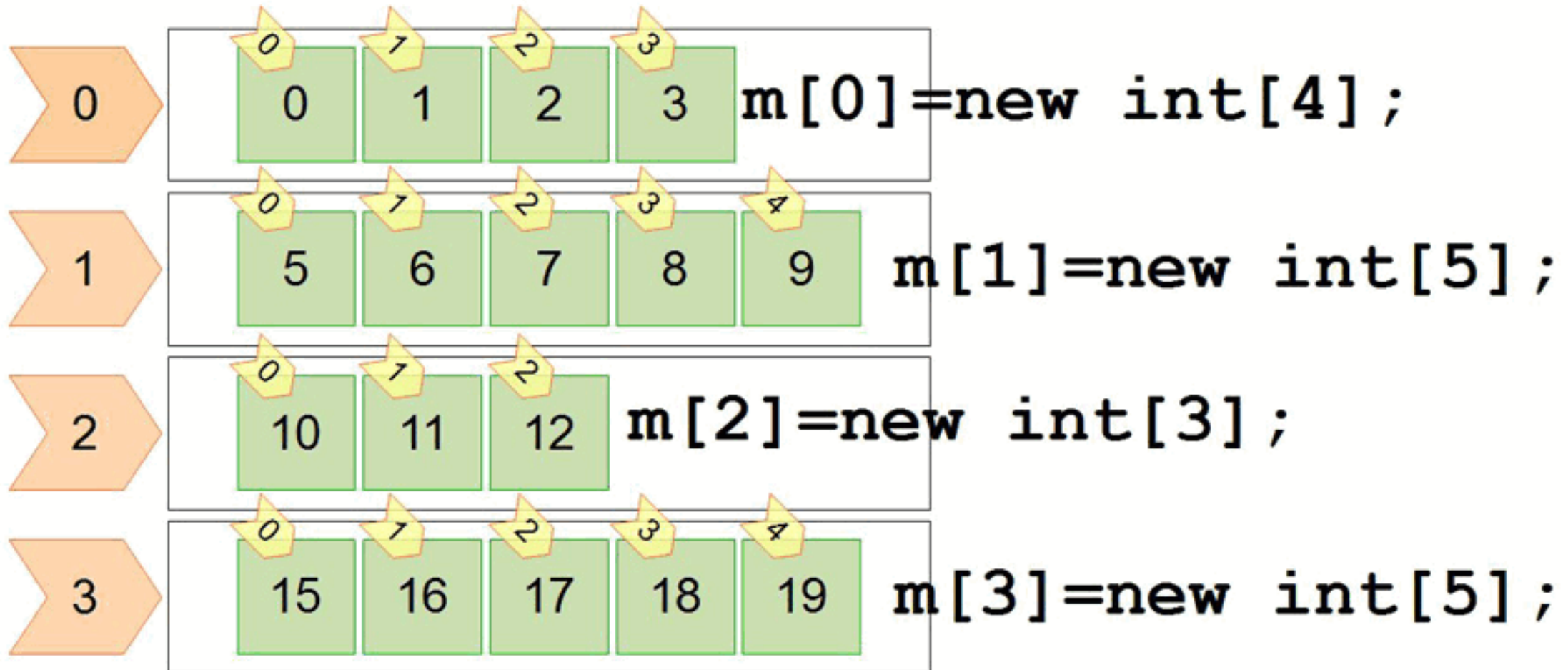
```
int suma = a2d[0][0] + a2d[0][1] + a2d[0][2] + a2d[0][3] + a2d[0][4];
```

Uso.

```
static int sumaarray2d(int[ ][ ] a2d){  
    int suma = 0;  
    for (int i1 = 0; i1 < a2d.length; i1++)  
        for (int i2 = 0; i2 < a2d[i1].length; i2++)  
            suma += a2d[i1][i2];  
    return suma;  
}
```

Uso.

```
int[][] m=new int[4][];
```



Inicialización.

¿En qué se diferencia la inicialización de arrays unidimensionales de arrays **multidimensionales**? En muy poco. La inicialización de los arrays multidimensionales es igual que la de los arrays unidimensionales.

```
int[ ][ ] inicializarArray (int n, int m)
{
    int[ ][ ] ret=new int[n][m];
    for (int i=0;i<n;i++)
        for (int j=0;j<m;j++)
            ret[i][j]=n*m;
    return ret;
}
```

Inicialización.

```
int[ ][ ] a2d={{0,1,2},{3,4,5},{6,7,8},{9,10,11}};
```

```
int[ ][ ][ ] a3d={{{0,1},{2,3}},{{0,1},{2,3}}};
```

4 filas x 3 columnas

Columnas

Filas

0	1	2
3	4	5
6	7	8
9	10	11

A2d[0][0]=0

A2d[3][0]=9

A2d[0][1]=1

A2d[3][1]=10

A2d[0][2]=2

A2d[3][2]=11

1ra regla de la programación: Si funciona no lo toques



Arrays multidimensionales.

Ejemplo

//Se crea una matriz de 4 filas y 3 columnas

```
int[][] a2d = new int [4][3];
```

//Se carga la matriz

```
System.out.println("Carga de valores en la matriz: ");  
    for (i = 0; i < 4; i++) { //Recorrido de filas  
        for (j = 0; j < 3; j++) { // Recorrido de columnas  
            System.out.print("a2d[" + i + "][" + j + "]= ");  
            a2d[i][j] = sc.nextInt();  
        }  
    }
```

Arrays multidimensionales.

Ejemplo

//Mostrar por pantalla los valores que contiene la matriz

```
System.out.println("valores introducidos:");  
for (i = 0; i < a2d.length; i++) {  
    for (j = 0; j < a2d[i].length; j++) {  
        System.out.print(a2d[i][j] + " ");  
    }  
    System.out.println(); //por cada fila un  
    Ln  
}
```

Arrays multidimensionales.

Ejemplo

//Calcular el máximo y mínimo

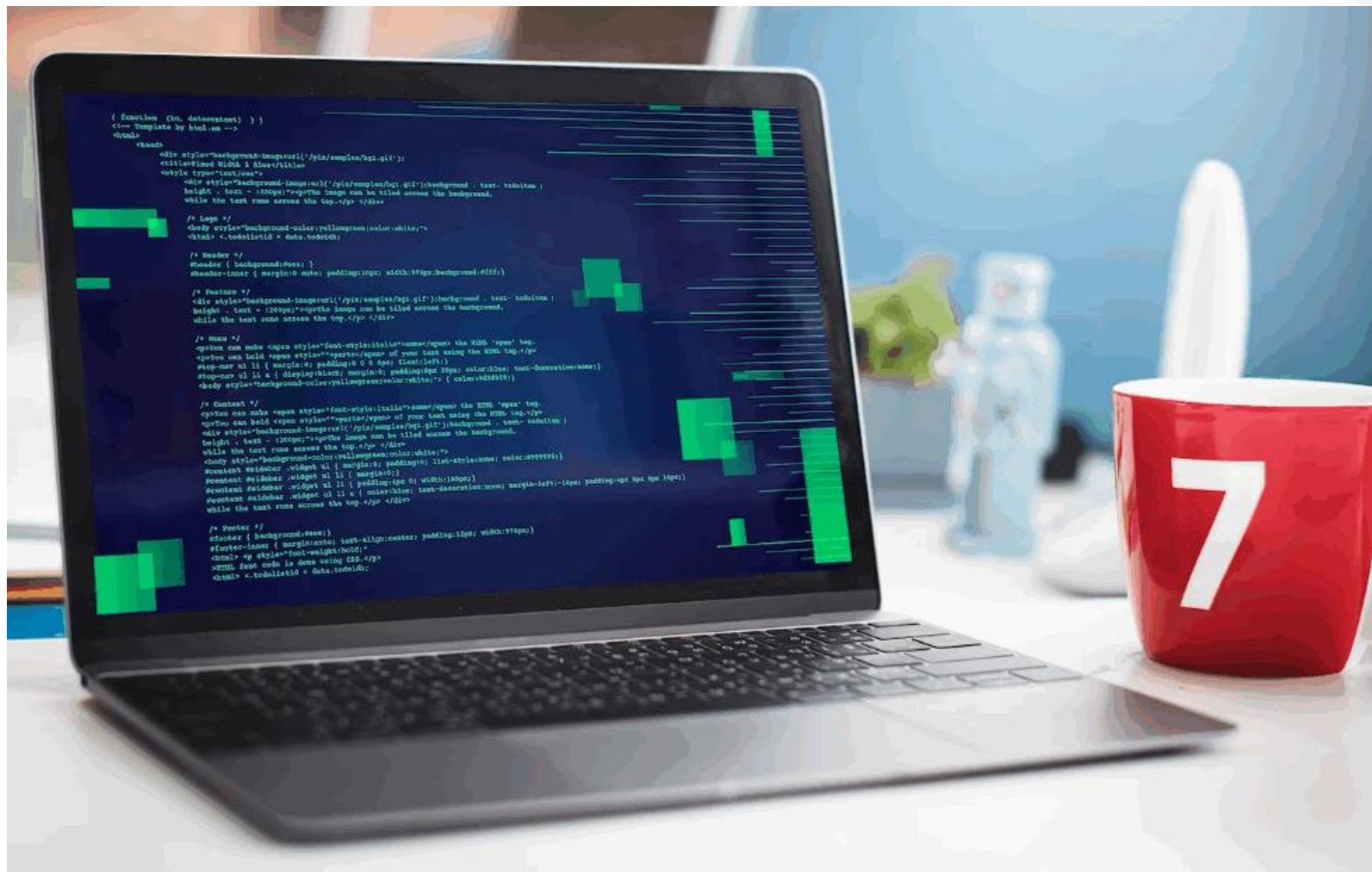
```
for (int x=0;x<matriz.length;x++){  
    for (int y=0;y<matriz[x].length;y++){  
        if (matriz[x][y] > numeroMayor){  
            numeroMayor = matriz[x][y];  
            mayorPosicion = x + "," + y;  
        }  
        if (matriz[x][y] < numeroMenor){  
            numeroMenor = matriz[x][y];  
            menorPosicion = x + "," + y;  
        }  
    }  
}
```

Arrays multidimensionales.

Ejemplo

```
ni Lectura de elementos de la matriz:
A[0][0]= 0
A[0][1]= 1
A[0][2]= 2
A[0][3]= 3
A[1][0]= 4
A[1][1]= 5
A[1][2]= 6
A[1][3]= 7
n2 A[2][0]= 8
A[2][1]= 9
A[2][2]= 10
A[2][3]= 11
A[3][0]= 12
A[3][1]= 13
A[3][2]= 14
A[3][3]= 15
A[4][0]= 16
A[4][1]= 17
A[4][2]= 18
A[4][3]= 19
valores introducidos:
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
16 17 18 19
Elemento mayor: 19 Fila: 4 Columna: 3
Elemento menor: 0 Fila: 0 Columna: 0
```


Ejercicios



Arrays multidimensionales. Ejercicios

1. Ordenar de mayor a menor la primera fila y mostrarla.
2. Ordenar de menor a mayor la segunda columna y mostrarla.
3. Calcular el máximo de la suma de los valores de cada fila, mostrar dichos valores de esa fila junto a la sumatoria.
4. Realizar una matriz de 4x3 de tipo String, cargar y mostrar el orden inverso que la ha añadido.