

# [EDA] Laboratorio 1

## 1 Ejercicio de desarrollo

Diseñe una solución algorítmica para el siguiente problema y analice el tiempo de ejecución utilizando notación  $\mathcal{O}f(n)$ .

### 1.1 Pares entre 2 filas

Diseñe un programa que, dada una matriz de tamaño  $M \times N$ , encuentre todos los pares de números pares que están en dos filas diferentes de la matriz.

Input:

- Dos números enteros  $M$  y  $N$  separados por espacio que representan las dimensiones de la matriz.
- Las siguientes  $M$  líneas contienen una secuencia de  $N$  números separados por espacios, que representan los elementos de la matriz.
- La siguiente línea contiene dos enteros,  $R1$  y  $R2$ , separados por espacio que representan las filas a procesar.

Output:

La salida posee dos filas, la primera contiene los valores de los números pares encontrados en la primera fila analizada de la matriz separados por espacio, la segunda posee también los números pares encontrados de la segunda fila separados por espacio.

Restricciones:

- $M$  y  $N$  pueden ser un valor entre 1 y 1000 inclusive.
- Los elementos de la matriz pueden ser un valor entre  $-10^8$  y  $10^8$
- Los valores de  $R1$  y  $R2$  pueden ser entre 1 y menor o iguales a  $M$ .

## 2 Ejercicios de análisis

Caracterice el tiempo de ejecución de los siguientes métodos en Java utilizando notación  $\mathcal{O}f(n)$ . Argumente su solución en no más de 5 líneas.

### 2.1 Invertir un número

```
1  import java.util.Scanner;
2
3  public class InvertirNumeroDecimal {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          System.out.print("Ingresa un numero decimal: ");
7          float numero = scanner.nextFloat();
8      }
```

```

9      float numeroInvertido = invertirDecimal(numero);
10
11      System.out.println("Numero invertido: " + numeroInvertido);
12  }
13
14  public static float invertirDecimal(float numero) {
15      int parteEntera = (int) numero;
16      float parteDecimal = numero - parteEntera;
17
18      int numeroInvertidoEntero = invertirEntero(parteEntera);
19      float numeroInvertidoDecimal = invertirDecimalParteDecimal(
20  parteDecimal);
21
22      return numeroInvertidoEntero + numeroInvertidoDecimal;
23  }
24
25  public static int invertirEntero(int numero) {
26      int numeroInvertido = 0;
27      while (numero != 0) {
28          numeroInvertido = numeroInvertido * 10 + numero % 10;
29          numero /= 10;
30      }
31      return numeroInvertido;
32  }
33
34  public static float invertirDecimalParteDecimal(float numero) {
35      float numeroInvertido = 0;
36      float factor = 1;
37      while (numero != 0) {
38          factor /= 10;
39          numeroInvertido += (numero % 10) * factor;
40          numero *= 10;
41          numero = (int)numero;
42      }
43      return numeroInvertido;
44  }
45  }

```

## 2.2 Multiplicación de matrices

```

1      import java.util.Scanner;
2
3      public class MultiplicacionMatrices {
4          public static void main(String[] args) {
5              Scanner scanner = new Scanner(System.in);
6
7              int filasA = scanner.nextInt();
8              int columnasA = scanner.nextInt();
9              int [][] matrizA = new int[filasA][columnasA];
10             ingresarMatriz(scanner, matrizA);
11
12             int filasB = scanner.nextInt();
13             int columnasB = scanner.nextInt();
14             int [][] matrizB = new int[filasB][columnasB];
15             ingresarMatriz(scanner, matrizB);

```

```

16
17     if (columnasA != filasB) {
18         return;
19     }
20
21     int[][] resultado = multiplicarMatrices(matrizA, matrizB);
22
23     imprimirMatriz(resultado);
24 }
25
26 public static void ingresarMatriz(Scanner scanner, int[][] matriz) {
27     for (int i = 0; i < matriz.length; i++) {
28         for (int j = 0; j < matriz[0].length; j++) {
29             matriz[i][j] = scanner.nextInt();
30         }
31     }
32 }
33
34 public static int[][] multiplicarMatrices(int[][] matrizA, int[][]
matrizB) {
35     int filasA = matrizA.length;
36     int columnasA = matrizA[0].length;
37     int filasB = matrizB.length;
38     int columnasB = matrizB[0].length;
39
40     int[][] resultado = new int[filasA][columnasB];
41
42     for (int i = 0; i < filasA; i++) {
43         for (int j = 0; j < columnasB; j++) {
44             for (int k = 0; k < columnasA; k++) {
45                 resultado[i][j] += matrizA[i][k] * matrizB[k][j];
46             }
47         }
48     }
49
50     return resultado;
51 }
52
53 public static void imprimirMatriz(int[][] matriz) {
54     for (int[] fila : matriz) {
55         for (int elemento : fila) {
56             System.out.print(elemento + " ");
57         }
58         System.out.println();
59     }
60 }
61 }

```

## Aspectos Formales

DE ACUERDO AL ARTÍCULO 50 DEL REGLAMENTO DE ESTUDIANTES DE PREGRADO

*“Cualquier conducta de un o una estudiante que vicie o tienda a viciar una actividad o evaluación académica, ya sea que se ejecute antes, durante o luego de su realización, dará origen a una o más de las siguientes sanciones, según la gravedad de la falta cometida: nota mínima (1,0) en la respectiva evaluación; reprobación*

*del curso respectivo; suspensión por un semestre o año académico, o expulsión de la Universidad.”*