

Programación I - 2^{do} semestre de 2022
Entregable: Complejidad Computacional

Fecha de entrega: Verificar en la página de la comisión correspondiente.

A continuación se plantea una serie de métodos escritos en Java. Para cada uno de ellos se pide indicar su orden de complejidad justificando el mismo con palabras y/o lenguaje matemático. La entrega es de carácter individual.

En cada ejercicio se debe indicar y justificar cuál es: la entrada, el tamaño de la entrada, la función de complejidad y el orden de complejidad del algoritmo. Se asume que todos los enteros pasados como parámetros son positivos y los arreglos, no vacíos.

1. El siguiente método calcula una suma relacionada con el producto de los primeros números naturales.

```
public static double calculo1(int n) {  
    double prod = 1.0;  
    double suma = 0.0;  
    for (int i = 0; i < n; i++) {  
        prod = 1.0;  
        for (int j = 1; j <= i; j++) {  
            prod = prod * j;  
        }  
        if (i % 2 == 0) {  
            suma += 1 / prod;  
        } else {  
            suma += -1 / prod;  
        }  
    }  
    return suma;  
}
```

2. Escribir **public static double calculo2(int n)** que realiza el mismo cálculo que el método **calculo1** pero con un orden de complejidad de menor. Justificar el mismo con palabras y/o lenguaje matemático.
3. El siguiente método construye una matriz cuadrada y anota en la diagonal central los elementos del arreglo pasado por parámetro, y en la diagonal superior anota números 1.

```
public static int[][] matrizDiag(int[] a) {  
    int n = a.length;  
    int b[][] = new int[n][n];  
    for (int i = 0; i < n; i++) {  
        for (int k = 0; k < n; k++) {  
            if (k == i) {  
                b[i][k] = a[i];  
            } else if (k == i+1) {  
                b[i][k] = 1;  
            } else {  
                b[i][k] = 0;  
            }  
        }  
    }  
    return b;  
}
```

4. Escribir un método **public static int[][]** `matrizDiag2(int[] a)` que devuelva exactamente lo mismo que el método `matrizDiag` pero con un orden de complejidad menor. Justificar el mismo con palabras y/o lenguaje matemático.
5. El siguiente método usa bloques de código de los cuales solo conocemos su complejidad.

```

public static void calculo3 (int n) {
    int m = 1;
    int i = 0;
    int suma = 0;
    while (m <= n) {
        m = m*2;
        i++;
    }
    System.out.println("Son " + i + " iteraciones del primer while.");

    while ( i <= n ) {
        suma += i * i;
        i++;
    }

    while ( i < n+100) {
        /* bloque de código de orden  $O(n^2)$  */
        i++;
    }

    while ( i < n * n) {
        /* bloque de código de orden  $O(1)$  */
        i++;
    }

    System.out.println("La suma es " + suma);
}

```

6. El siguiente método recursivo aproxima a la raíz cuadrada de 2.

```

public static double raiz2Aprox(int n) {
    if (n == 0) {
        return 1;
    }
    return raiz2Aprox(n-1) / 2 + 1 / raiz2Aprox(n-1);
}

```