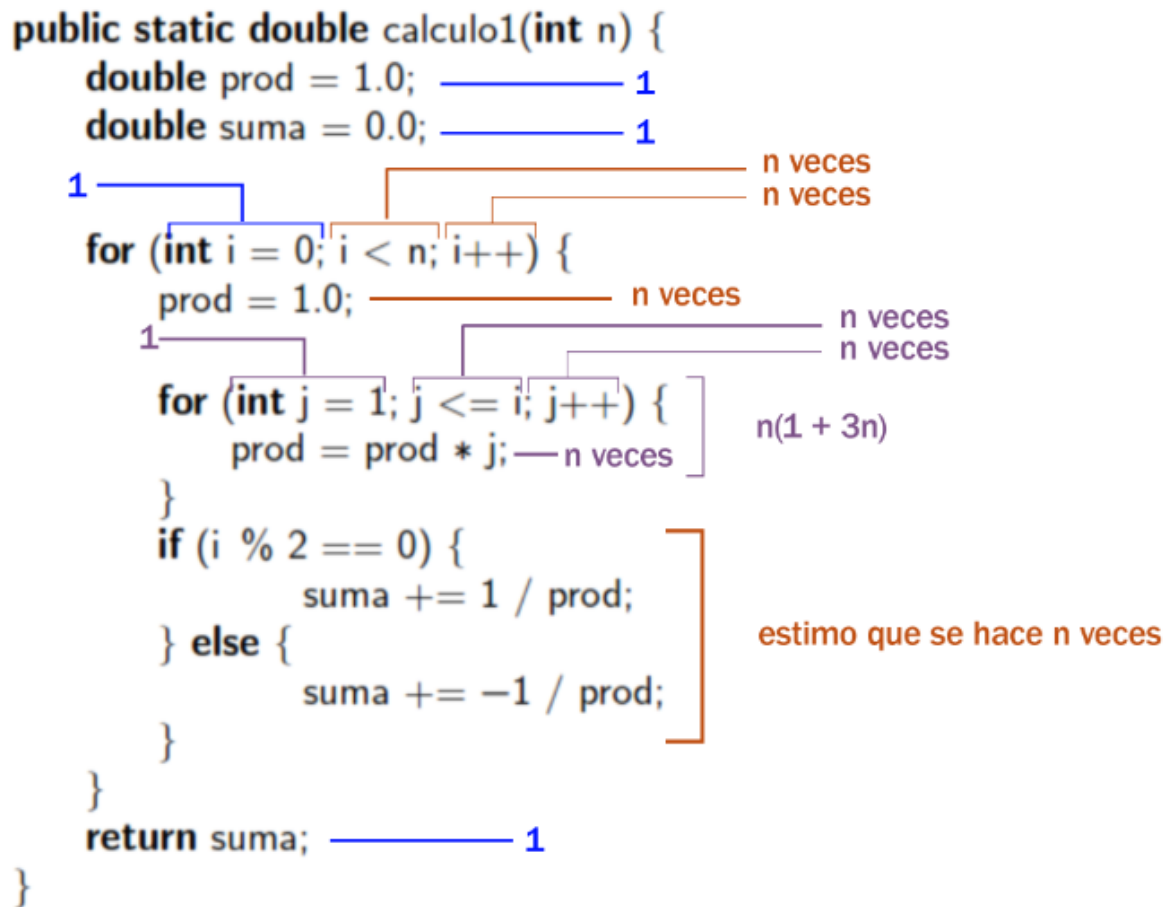


1- Ejercicio 1



Entrada = n ;

Tamaño de entrada = n ;

Función de complejidad = $F(n) = 4 + 4n + n(1+3n) = 4+4n+n+ 3n^2 = 3n^2 + 5n + 4$

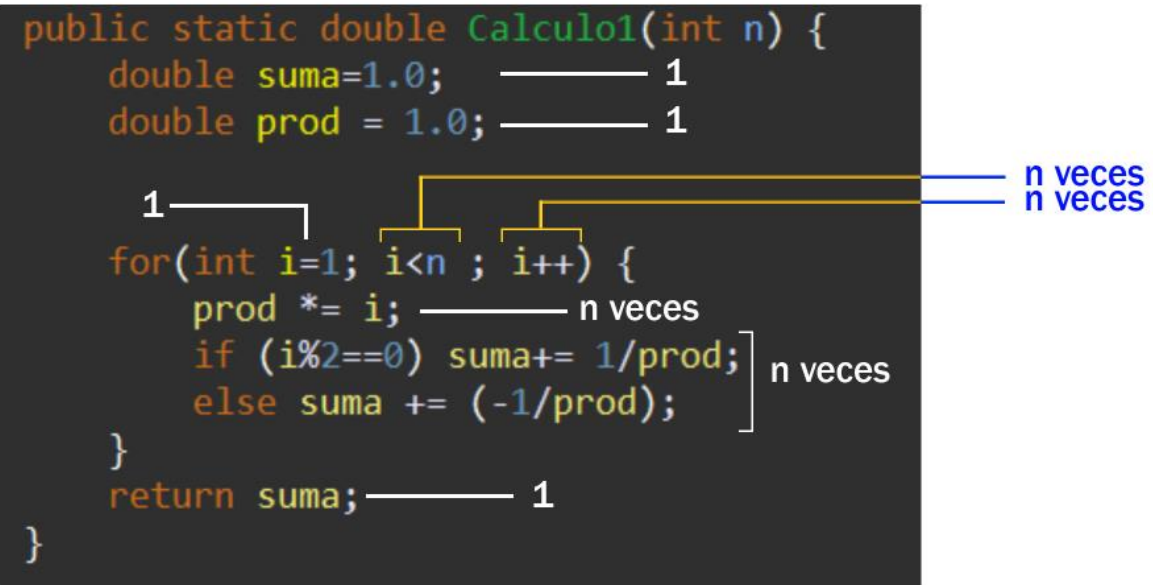
Orden de complejidad = $F \in O(n^2)$

$F(n) \leq \alpha n^2 \quad \forall n \geq n^2$

$3n^2 + 5n + 4 \leq 12n^2$

2- Ejercicio 2

```
public static double Calculo1(int n) {  
    double suma=1.0;  _____ 1  
    double prod = 1.0; _____ 1  
  
    1 _____  
    for(int i=1; i<n ; i++) {  
        prod *= i; _____ n veces  
        if (i%2==0) suma+= 1/prod; ] n veces  
        else suma += (-1/prod);  
    }  
    return suma; _____ 1  
}
```



Entrada = n ;

Tamaño de entrada = n ;

Función de complejidad = $F(n) = 4 + 4n = 4 + 4n = 4n + 4$

Orden de complejidad = $F \in O(n)$

$F(n) \leq \alpha n \quad \forall n \geq n_0$

$4n + 4 \leq 8n$

3- Ejercicio 3

```

public static int[][] matrizDiag(int[] a) {
    int n = a.length; 1
    int b[][] = new int[n][n]; 1
    1 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; 1
}

```

Diagrama de complejidad:

- La línea `int n = a.length;` tiene una complejidad de 1.
- La línea `int b[][] = new int[n][n];` tiene una complejidad de 1.
- El bucle `for (int i = 0; i < n; i++)` se ejecuta n veces.
- Dentro del bucle `i`, el bucle `for (int k = 0; k < n; k++)` se ejecuta n veces.
- El bloque de código dentro del bucle `k` (las líneas `if`, `else if` y `else`) se ejecuta n veces.
- El bloque de código dentro del bucle `k` se repite n veces para cada `i`, lo que resulta en una complejidad de $n(1 + 3n)$.
- La línea `return b;` tiene una complejidad de 1.

Entrada = `a`;

Tamaño de entrada = `a.length` = n ;

Función de complejidad = $F(n) = 4 + 2n + n(1+3n) = 4+2n+n+ 3n^2 = 3n^2 + 3n + 4$

Orden de complejidad = $F \in O(n^2)$

$F(n) \leq \alpha n^2 \quad \forall n \geq n^2$

$3n^2 + 3n + 4 \leq 10n^2$

4- Ejercicio 4

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

Entrada = **a**;

Tamaño de entrada = **a.length = n**;

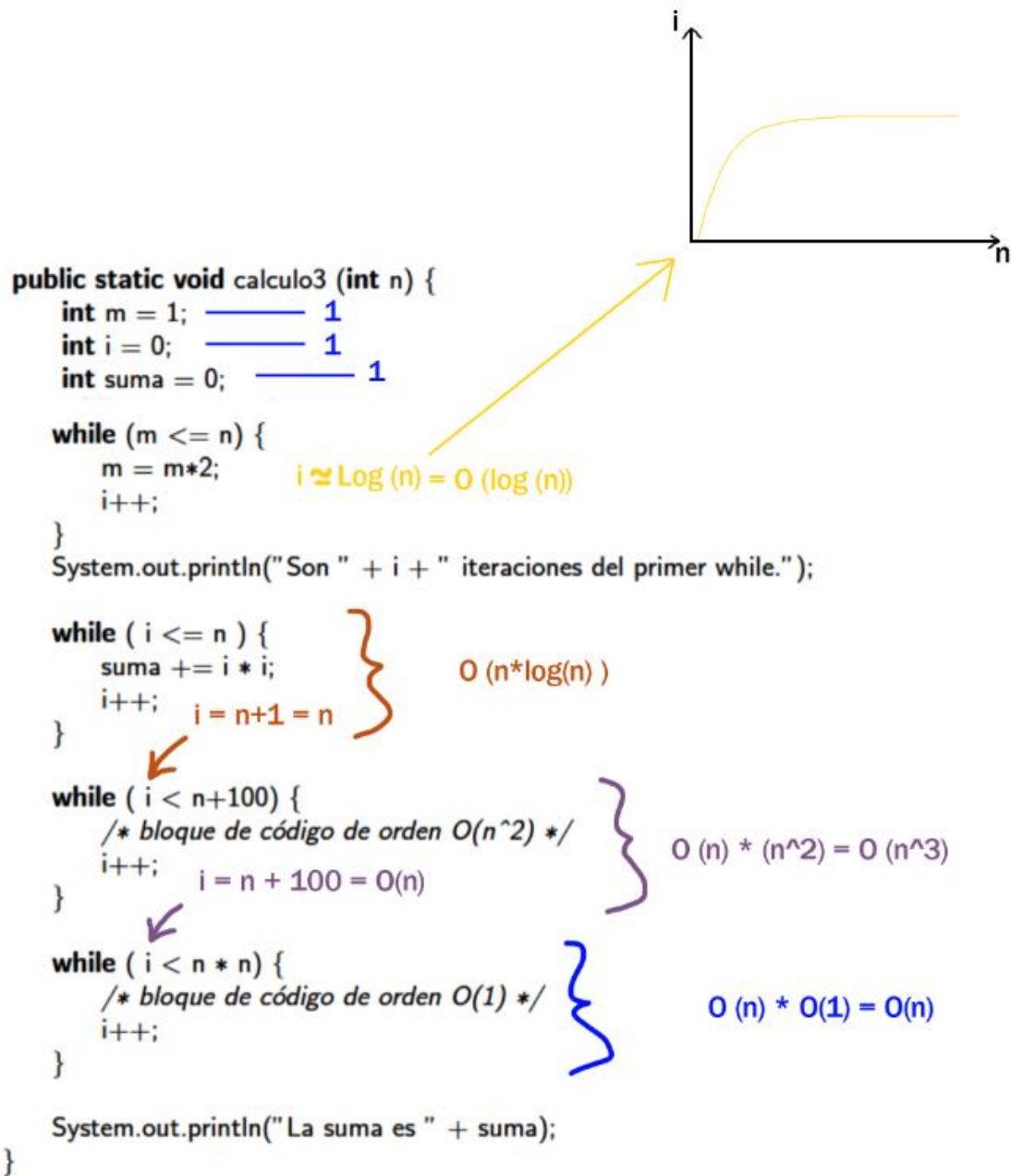
Función de complejidad = $F(n) = 5 + 4n = 4n + 5$

Orden de complejidad = **$F \in O(n)$**

$F(n) \leq \alpha n \quad \forall n \geq n_0$

$4n + 5 \leq 9n$

5- Ejercicio 5



Entrada = n ;

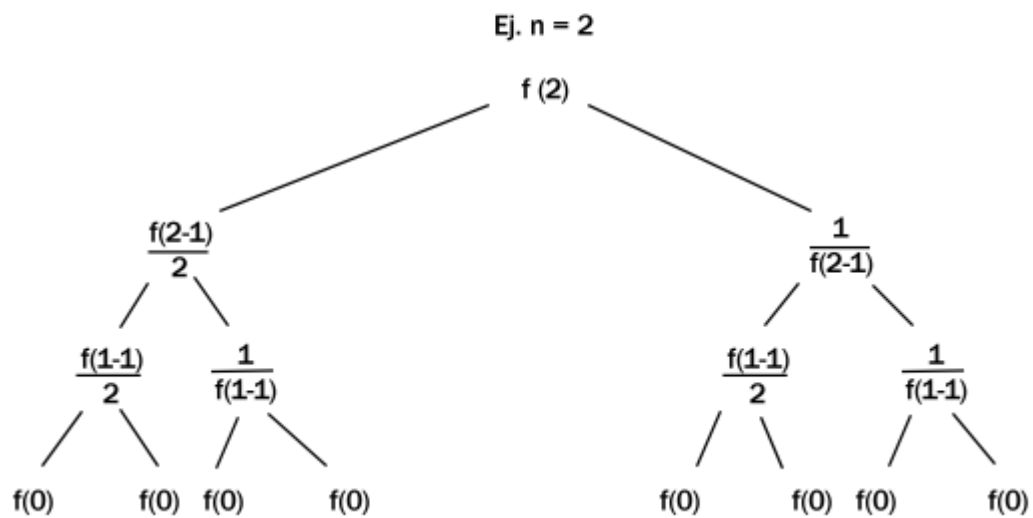
Tamaño de entrada = n ;

Función de complejidad = $F(n) = O(1) + O(\log(n)) + O(n * \log(n)) + O(n^3) + O(n) =$

$O(n^3) \Rightarrow$ peor de los casos

6- Ejercicio 6

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



Función de complejidad = $O(2^n)$