Asignación 10 – Depuración de recursividad

Nombre:

Jose Miguel Rojo Cota

ID:

00000267850

Maestro:

Francisco Antonio Mejía Domínguez

Materia:

Estructuras de Datos

Fecha:

22/10/2025

Depuración Manual de Recursividad

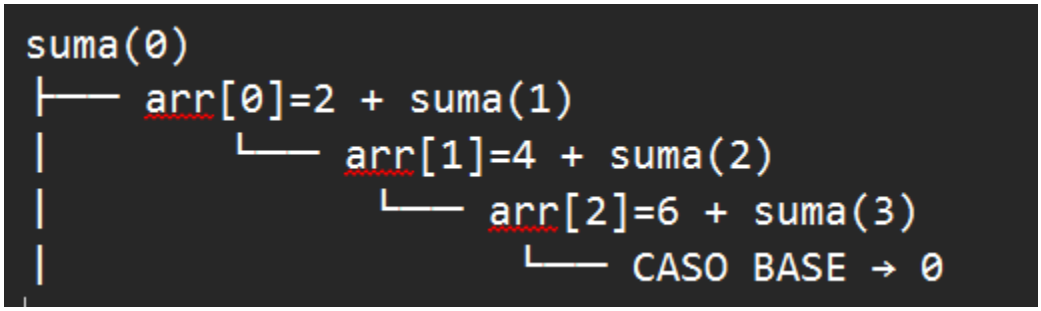
Ejercicio 1: sumaRecursiva

Código:

```java
public static int sumaRecursiva(int[] arr, int index) {
    // CASO BASE: si el indice esta fuera de rango
    if (index >= arr.length) {
        return 0;
    }
    // CASO RECURSIVO: elemento actual + suma del resto
    return arr[index] + sumaRecursiva(arr, index + 1);
}
```

Ejemplo: arr = [2, 4, 6]

Árbol de Recursión:



Traza paso a paso:

FASE 1: Bajando (llamadas)
1 sumaRecursiva([2,4,6], 0)
2 sumaRecursiva([2,4,6], 1)
3 sumaRecursiva([2,4,6], 2)
4 sumaRecursiva([2,4,6], 3) → caso base → retorna 0

FASE 2: Retornando (resultados)
5 suma(2) calcula: 6 + 0 = 6 → retorna 6
6 suma(1) calcula: 4 + 6 = 10 → retorna 10
7 suma(0) calcula: 2 + 10 = 12 → retorna 12

Resultado final: 12

Pila de ejecución:

Momento de máxima profundidad:
sumaRecursiva([2,4,6], 3) ← tope
sumaRecursiva([2,4,6], 2)
sumaRecursiva([2,4,6], 1)
sumaRecursiva([2,4,6], 0) ← base

Cuando retorna 0:
sumaRecursiva([2,4,6], 2)
sumaRecursiva([2,4,6], 1)
sumaRecursiva([2,4,6], 0)

Pila vacía → resultado final = 12

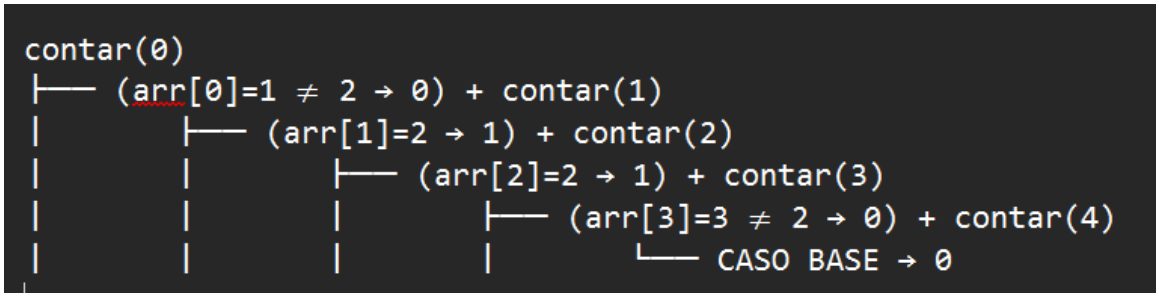
Ejercicio 2: contarRecursivo

Código:

```java
public static int contarRecursivo(int[] arr, int x, int index) {
    // CASO BASE: indice fuera de rango
    if (index >= arr.length) {
        return 0;
    }
    // CASO RECURSIVO: suma 1 si coincide, 0 si no + recursion en el resto
    int coincide = (arr[index] == x) ? 1 : 0;
    return coincide + contarRecursivo(arr, x, index + 1);
}
```

Ejemplo: arr = [1, 2, 2, 3], x = 2

Árbol de Recursión:



Traza paso a paso:

FASE 1: Bajando
1 contar([1,2,2,3],2,0)
2 contar([1,2,2,3],2,1)
3 contar([1,2,2,3],2,2)
4 contar([1,2,2,3],2,3)
5 contar([1,2,2,3],2,4) → caso base → retorna 0

FASE 2: Retornando
6 contar(3) calcula: 0 + 0 = 0 → retorna 0
7 contar(2) calcula: 1 + 0 = 1 → retorna 1
8 contar(1) calcula: 1 + 1 = 2 → retorna 2
9 contar(0) calcula: 0 + 2 = 2 → retorna 2

Resultado final: 2

Pila de ejecución:

Máxima profundidad:
contar([1,2,2,3],2,4) ← tope
contar([1,2,2,3],2,3)
contar([1,2,2,3],2,2)
contar([1,2,2,3],2,1)
contar([1,2,2,3],2,0) ← base

Después de resolver casos base:
contar([1,2,2,3],2,2)
contar([1,2,2,3],2,1)
contar([1,2,2,3],2,0)

Pila vacía → resultado final = 2