

Juan Pablo Yepiz Leon - 268712

Jueves
23/10/2025

Tarea: Depuración Manual de Recursión

Código

```
public static int sumaRecursiva(int[] arr, int index) {  
    // Caso base: índice fuera del arreglo  
    if (index == arr.length) return 0;  
    // Caso recursivo: elemento actual + suma del resto  
    return arr[index] + sumaRecursiva(arr, index + 1);  
}
```

Árbol de recursión para $\text{sumaRecursiva}(\{4, 2, 7, 1\}, 0)$

$\text{sumaRecursiva}(0)$

$4 + \text{sumaRecursiva}(1)$

$2 + \text{sumaRecursiva}(2)$

$7 + \text{sumaRecursiva}(3)$

$1 + \text{sumaRecursiva}(4)$

$0 \leftarrow$ Caso base

Trazo de ejecución paso a paso

Fase 1: Bajando (haciendo llamadas)

Paso 1: Llamada $\rightarrow \text{sumaRecursiva}(\text{arr}, 0)$

Paso 2: Llamada $\rightarrow \text{sumaRecursiva}(\text{arr}, 1)$

Paso 3: Llamada $\rightarrow \text{sumaRecursiva}(\text{arr}, 2)$

Paso 4: Llamada $\rightarrow \text{sumaRecursiva}(\text{arr}, 3)$

Paso 5: Llamada $\rightarrow \text{sumaRecursiva}(\text{arr}, 4)$ (Caso base)

Fase 2: Retomando (obteniendo resultados)

Paso 6: SumaRecursiva(4) retorna 0 (caso base)

Paso 7: SumaRecursiva(3) calcula $1+0=1$ y retorna 1

Paso 8: SumaRecursiva(2) calcula $7+1=8$ y retorna 8

Paso 9: SumaRecursiva(1) calcula $2+8=10$ y retorna 10

Paso 10: SumaRecursiva(0) calcula $4+10=14$ y retorna 14

Resultado Final: 14

Pila de ejecución (Call Stack)

Momento 1: Máxima Profundidad (antes del caso base)

Pila:

SumaRecursiva(Carr, 4) ←

Tope

SumaRecursiva(Carr, 3)

SumaRecursiva(Carr, 2)

SumaRecursiva(Carr, 1)

SumaRecursiva(Carr, 0) ←

Base

La función ha bajado hasta $\text{index} = 4$, lista para retornar 0.

Momento 2: Subiendo (retornando valores)

SumaRecursiva(Carr, 3) ←

Tope

SumaRecursiva(Carr, 2)

SumaRecursiva(Carr, 1)

SumaRecursiva(Carr, 0) ←

Base

Después de que SumaRecursiva(Carr, 4) retorna 0:

SumaRecursiva(Carr, 3) calcula $1+0=1$

Se desapila SumaRecursiva(Carr, 3)

Momento 3: Al finalizar todo

[Pila vacía]

Resultado obtenido: 14

Código

```
Public static String invertirRekursiva(String S) {  
    // Caso base: longitud 0 o 1  
    if (S.length() <= 1) return S;  
    // Caso recursivo: último carácter + invertir resto  
    return S.charAt(S.length() - 1) + invertirRekursiva(  
        S.substring(0, S.length() - 1));  
}
```

Árbol de recursión para invertirRekursiva("Hola")

invertirRekursiva("Hola")
 |
 'a' + invertirRekursiva("Hol")
 |
 'l' + invertirRekursiva("Ho")
 |
 'o' + invertirRekursiva("H")
 |
 "H" ← Caso base

Trazo de ejecución paso a paso

Fase 1: bajando llamadas recursivas)

Paso 1: Llamada → invertirRekursiva("Hola")

Paso 2: Llamada → invertirRekursiva("Hol")

Paso 3: Llamada → invertirRekursiva("Ho")

Paso 4: Llamada → invertirRekursiva("H")

Fase 2: Retornando (resolviendo valores)

Paso 5: invertirRekursiva("H") retorna "H" (caso base)

Paso 6: invertirRekursiva("Ho") calcula 'o' + "H" y retorna "oH"

Paso 7: invertirRecursiva("Ho") Calcula 'l' + "oH" y
retorna "loH"

Paso 8: invertirRecursiva("Holo") calcula 'a' + "loH" y
retorna "aloH"

Resultado Final: "aloH"

Pila de ejecución (Call Stack)

Momento 1: Máxima Profundidad (antes del caso base)

Pila:	La función ha bajado
invertirRecursiva("H") ←	hasta la cadena "H"
Tope	lista para retornar
invertirRecursiva("Ho")	"H".
invertirRecursiva("Hol")	
invertirRecursiva("Holo") ←	
Base	

Momento 2: Subiendo (retornando valores)

invertirRecursiva("Ho") ←	Después de que invertir
Tope	Recursiva("H") retorna "H"
invertirRecursiva("Hol")	
invertirRecursiva("Holo") ←	invertirRecursiva("Ho")
Base	Calcula "oH"

Momento 3: Al finalizar todo
[Pila vacía] Luego se desapila al
terminar

Resultado obtenido: "aloH"