



Misiones, República Argentina

INTRODUCCIÓN A LA PROGRAMACIÓN

ORDENAMIENTO DE BURBUJA

CLASE 10

ORDENAMIENTO

- Ordenar una estructura de datos consiste en reacomodar sus elementos de acuerdo a algún **criterio**.

Por ejemplo, los mensajes pueden ordenarse por asunto, por fecha, por contacto.

- Si varios elementos coinciden de acuerdo al criterio, pueden definirse **niveles de criterios**.

Por ejemplo, los mensajes pueden ordenarse por asunto y dentro de un mismo asunto, cronológicamente por fecha.

- Existen distintos algoritmos de ordenamiento, que pueden describirse independientemente del problema.

ORDENAMIENTO: Método de la Burbuja

- El método de ordenamiento burbuja consiste en comparar cada elemento de la estructura con el siguiente e intercambiándolos si corresponde.
- El proceso se repite hasta que la estructura esté ordenada.



- El orden se establece de acuerdo a la clave y la estructura tiene que tener acceso directo a sus componentes.

ALGORITMO DE LA BURBUJA

Consiste en comparar si el primer elemento del vector es mayor que el segundo, si esta condición se cumple entonces una variable temporal toma el valor del primer elemento del vector y el primer elemento del vector toma la posición del segundo elemento por último el segundo elemento toma el valor de la variable temporal, logrando con este método pasar el valor de la primera posición a la segunda posición.

Comencemos con un ejercicio: Hacer el Algoritmo para ordenar de manera efectiva un Vector

1. Un vector estático.
2. Un vector cargado por teclado.
3. Mostrar el vector.

Algoritmo Burbuja

repetir mientras no este ordenada
para cada par de elementos adyacentes
si el elemento es mayor que el que sigue
intercambiar

Algoritmo Ordenamiento_Burbuja**Definir** vector, temporal **Como Entero****Dimension** vector(5)

Vector(1)=8

Vector(2)=1

Vector(3)=44

Vector(4)=4

Vector(5)=5

Para i<-1 **Hasta** 4 **Con Paso** 1 **Hacer** **Si** Vector(i) > vector(i+1) **Entonces**

temporal = vector(i)

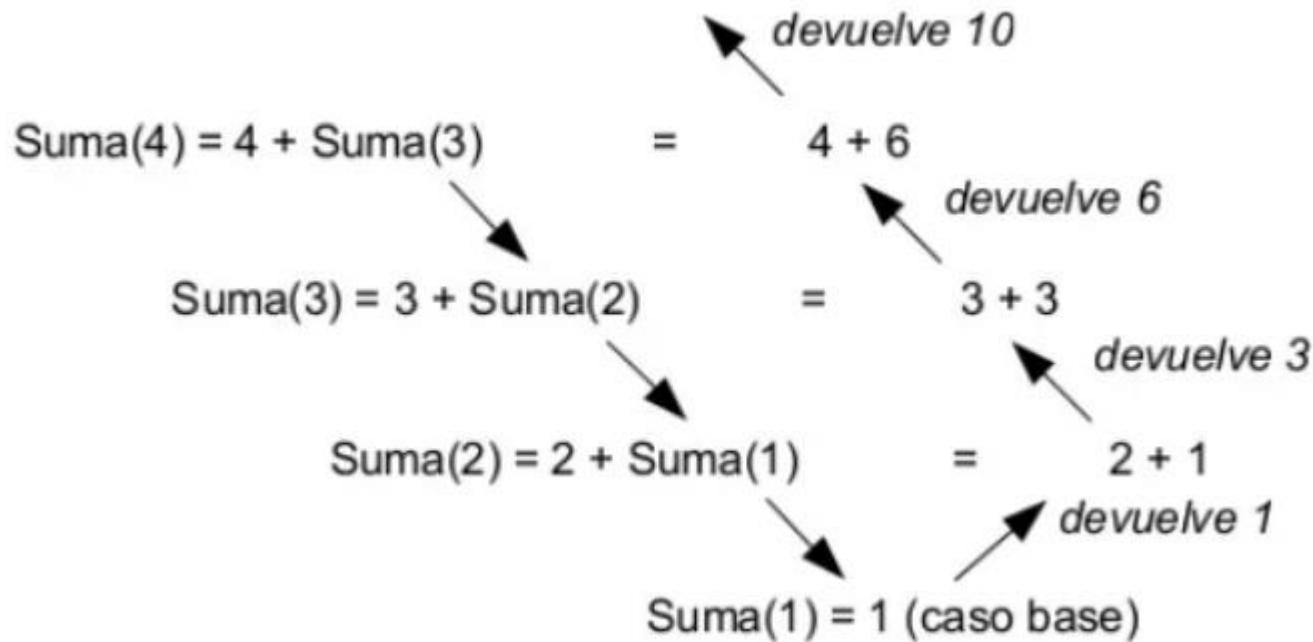
Vector(i)=Vector(i+1)

vector(i+1)=temporal

Fin Si **Fin Para****Para** i<-1 **Hasta** 5 **Con Paso** 1 **Hacer** **imprimir** vector(i) **Fin Para****FinAlgoritmo**

FUNCIÓN RECURSIVA

Una función recursiva es aquella que al ejecutarse hace llamadas a ella misma.

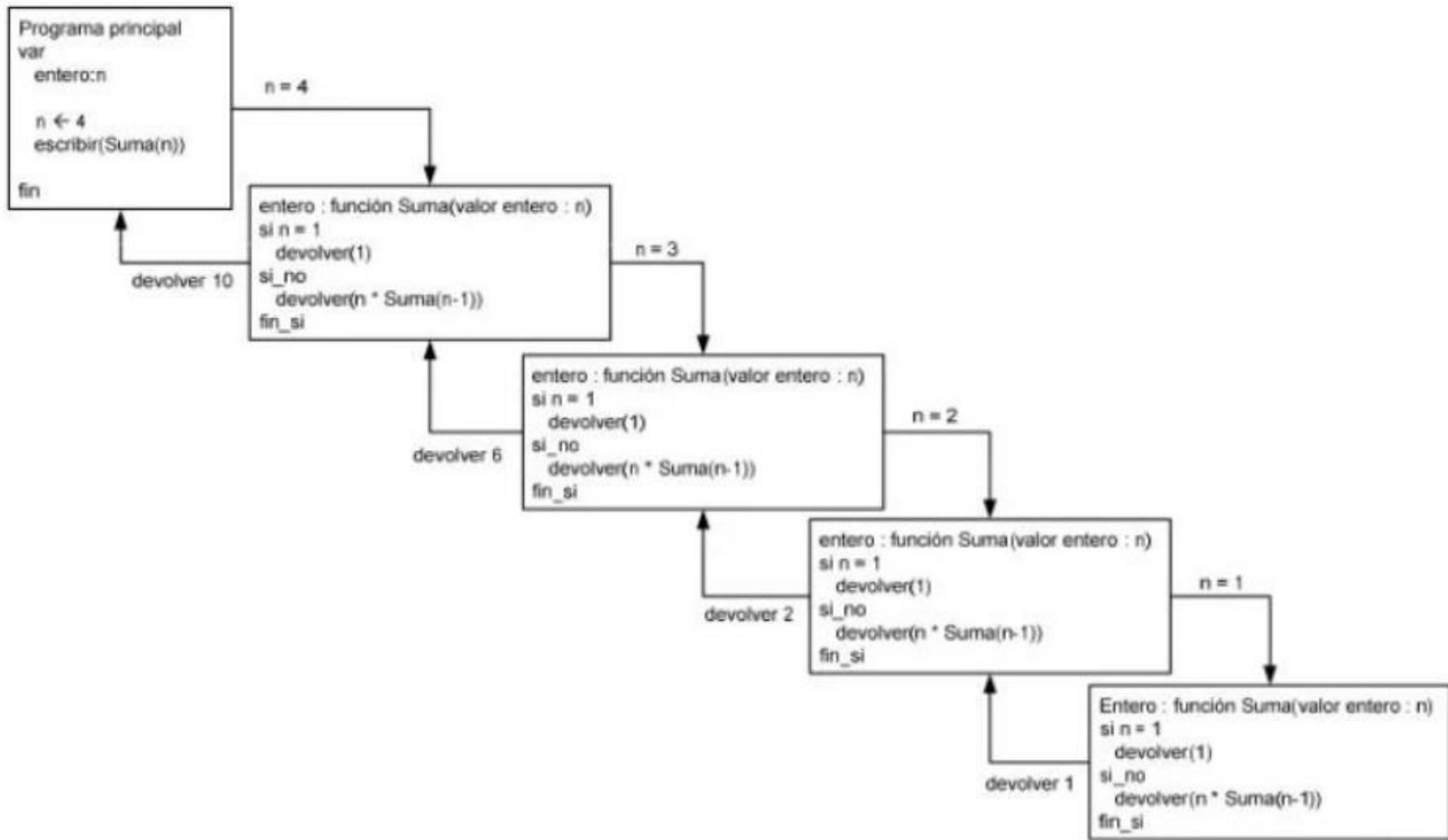


FUNCIÓN RECURSIVA

CASO RECURSIVO: Una solución que involucra volver a utilizar la función original, con parámetros que se acercan más al caso Base. Los pasos que sigue el caso recursivo son los siguientes:

1. El procedimiento se llama a sí mismo.
2. El problema se resuelve, resolviendo el mismo problema pero de tamaño menor.
3. La manera en la cual el tamaño del problema disminuye asegura que el caso base eventualmente se alcanzará.

FUNCIÓN RECURSIVA



Ejemplo:

```
Proceso numeroPrimoRecursivo|  
  
    Funcion val <- numeroPrimoRecursivo(numero, c)  
  
        Si ( numero mod c = 0 Y numero <> 2 ) Entonces  
  
            val <- falso  
  
        Sino Si (c > numero / 2) Entonces  
  
            val <- verdadero  
  
        Sino  
  
            val <- numeroPrimoRecursivo( numero, c+1 )  
  
        FinSi  
  
    FinSi  
  
FinFuncion  
  
Escribir "favor ingresar un numero: "  
Leer numero  
  
Si ( numeroPrimoRecursivo(numero, 2) ) Entonces  
    Escribir "El numero es primo"  
Sino  
    Escribir "El numero NO es primo"  
FinSi  
  
FinProceso
```

MUCHAS GRACIAS!!

Sigamos practicando...

