



## Semana 3: Recursividad Aplicada



**Datos** 

Materia: Estructura de Datos Aplicadas

Alumno: Eli Haziel Ortiz Ramírez

Grupo y Grado: 4 B DSM

Todos los códigos en:

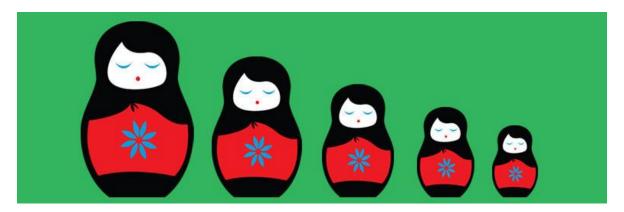


# Índice

Índice	2
Introducción	3
Actividad 3	4
Suma de Enteros Positivos (Suma Rusa):	4
Mínimo Común Divisor	5
Explicar programa	6

## Introducción

La recursividad, en términos simples, es un concepto fundamental en programación que implica resolver un problema a partir de la solución de subproblemas más pequeños y similares. A través de este informe, exploraremos ejercicios prácticos que utilizan este enfoque, permitiéndonos comprender su aplicabilidad en el ámbito de la programación y sus beneficios para abordar desafíos de manera efectiva y elegante.



#### Actividad 3

#### Suma de Enteros Positivos (Suma Rusa):

Descripción: Un programa que encuentre la suma de los enteros positivos pares desde N hasta 2, Corroborar que si N es impar se imprima un mensaje de error.

```
Estructura-de-Datos - suma-rusa.ts
    // Suma por el método Ruso usando Recursividad
           В
    // 14 26
4 // 7 52
5 // 3 104
6 // 1 208
7 // suma = 52 + 104 + 208
9 const sumaRusa = (A:number, B:number, suma = 0) \Rightarrow {
        let sumando = 0;
11
       if (A \le 1) {
            return suma;
        } else if (A % 2 \neq 0 ) {
            Math.floor(A);
            console.log(B);
            return (sumaRusa(A/2,B * 2, suma + B));
        } else {
            return (sumaRusa(A/2,B * 2, suma));
        }
22 }
    console.log(sumaRusa(27,82));
```

#### Mínimo Común Divisor

Escribir un programa que calcule el máximo común divisor (MCD) de dos enteros positivos. Si M >= N una función recursiva para MCD es

```
MCD = M si N = <> 0
```

 $MCD = MCD (N,M \mod N) \text{ si } N <> 0$ 

El programa le debe permitir al usuario ingresar los valores para M y N desde la consola. Una función recursiva es entonces llamada para calcular el MCD. El programa entonces imprime el valor para el MCD.

```
Estructura-de-Datos - mcd.ts

// Función para calcular el maximo común divisor dado 2 números

function max_comun_div(n1: number, n2: number, divisor = 2): number {

// Caso base: Si ambos números son 1, el max_comun_div es 1

if (n1 = 1 86 n2 == 1) {

return 1;

} else {

// Verificar si ambos números son divisibles por el divisor actual divisor

if (n1 % divisor == 0 86 n2 % divisor == 0) {

// Llamada recursiva dividiendo n y n1 por divisor y multiplicando el resultado por divisor return divisor * max_comun_div(n1 / divisor, n2 / divisor, divisor);

} else if (n1 % divisor == 0) {

// Llamada recursiva dividiendo n por divisor return max_comun_div(n1 / divisor, n2, divisor);

} else if (n2 % divisor == 0) {

// Llamada recursiva dividiendo n1 por divisor return max_comun_div(n1, n2 / divisor, divisor);

} else {

// Si ninguno de los casos anteriores se cumple, incrementar divisor y llamar recursivamente return max_comun_div(n1, n2, divisor + 1);

}

// Llamada a la función max_comun_div con los valores dados, e imprimir el resultado console.log(max_comun_div(40, 60));
```

### Explicar programa

```
Estructura-de-Datos - programa.ts

function f(x:number):number {

if (x > 100) {
    console.log(x);
    return x - 10;
    } else {
    console.log(x);
    return(f(f(x+11)));
}

console.log(f(91));
```

El programa es una función recursiva que se repite y llama a sí misma cada que x es menor a 100, en cuyo caso siempre retorna 91, valor que equivale siempre a la cantidad de llamadas que se hace así misma hasta retornar un valor. En caso de que el valor pasado sea mayor a 100, simplemente devuelve ese número menos 10, siempre y cuando sea mayor de 100.