



**Tecnológico
de Monterrey**

Act 1.1 - Funciones Iterativas, Recursivas y su análisis de Complejidad.

Daniel Esparza Arizpe - A01637076

Programación de estructuras de datos y algoritmos fundamentales

Grupo 605

Tecnológico de Monterrey

Domingo 20 de agosto 2023

Act 1.1 - Funciones Iterativas, Recursivas y su análisis de Complejidad.

Código:

```
int sumaIterativa(int n){
    int sumaI = 0;
    for (int i = 1; i <= n; i = i+1){
        sumaI += i;
    }
    return sumaI;
}

int sumaRecursiva(int n){
    if (n == 1){
        return 1;
    }
    else{
        return n + sumaRecursiva(n - 1);
    }
}

int sumaDirecta(int n){
    int sumaD = n*(n+1)/2;
    return sumaD;
}

int main(){
    int n;
    cout << "Ingrese un numero 'n': ";
    cin >> n;

    cout << "La suma de los numeros de 1 a " << n << " de forma iterativa es: " << sumaIterativa(n) << endl;
    cout << "La suma de los numeros de 1 a " << n << " de forma recursiva es: " << sumaRecursiva(n) << endl;
    cout << "La suma de los numeros de 1 a " << n << " de forma directa es: " << sumaDirecta(n) << endl;
    return 0;
}
```

Resultados con n = 10

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
● danielspazarizpe@MacBook-Pro-de-Daniel-2 tiposDeSumas % ./sumas.exe
Ingrese un numero 'n': 10
La suma de los numeros de 1 a 10 de forma iterativa es: 55
La suma de los numeros de 1 a 10 de forma recursiva es: 55
La suma de los numeros de 1 a 10 de forma directa es: 55
○ danielspazarizpe@MacBook-Pro-de-Daniel-2 tiposDeSumas %
```

Análisis de complejidad

La función **sumaIterativa** utiliza un bucle que itera 'n' veces, lo que resulta en una complejidad de $O(n)$. La función **sumaRecursiva** utiliza recursión en 'n' llamadas, también teniendo una complejidad de $O(n)$. En contraste, la función **sumaDirecta** utiliza una fórmula matemática constante sin importar el tamaño de 'n', lo que le otorga una complejidad de $O(1)$.