



#### **Ejercicio 4**

Docente: Jimmy Nataniel Requena Llorentty

Materia: Programación III

Carrera: Ingeniería En Sistemas

Estudiantes: Joaquin Marcos Maita Flores

Santa Cruz – Bolivia

2025

### Sobrecarga de sumar

```
#include <iostream> // Para std::cout, std::endl
#include <string>    // Para std::string

// Versión 1: Suma dos enteros
int sumar(int a, int b) {
    std::cout << "Ejecutando sumar(int, int)... ";
    return a + b;
}

// Versión 2: Suma dos números de punto flotante (double)
// ¡Sobrecargada! Mismo nombre, diferente tipo de parámetros.
double sumar(double a, double b) {
    std::cout << "Ejecutando sumar(double, double)... ";
    return a + b;
}

// Versión 3: Concatena dos cadenas (std::string)
// ¡Sobrecargada! Mismo nombre, diferente tipo de parámetros.
std::string sumar(const std::string& a, const std::string& b) {
    std::cout << "Ejecutando sumar(const std::string&, const std::string&)... ";
    return a + b;
}

// Versión 4: Suma tres enteros
// ¡Sobrecargada! Mismo nombre, diferente número de parámetros.
int sumar(int a, int b, int c) {
    std::cout << "Ejecutando sumar(int, int, int)... ";
    return a + b + c;
}

//Ahora como dijo daredevil "haber si sale"

int multiplicar (int x, int y){

    std::cout<<"\n1ro = ";
    return x * y;
}
int multiplicar (int x, int y, int z){
    std::cout<<"\n2do = ";
    return x * y * z;
}

int main() {
```

```
std::cout << "Suma de enteros (5, 3): " << sumar(5, 3) << std::endl;
std::cout << "Suma de doubles (5.5, 3.3): " << sumar(5.5, 3.3) << std::endl;
std::cout << "Concatenacion de strings (\\"Hola, \\", \\"Mundo!\"): "
    << sumar(std::string("Hola, "), std::string("Mundo!")) <<
std::endl;
std::cout << "Suma de tres enteros (1, 2, 3): " << sumar(1, 2, 3) <<
std::endl;
std::cout<<"\nejecutando no una suma si no una multiplicacion por que es lo
unico que se me ocurrio ojala este bien el ejercicio\n"<<std::endl;

std::cout << "\n1ro" << multiplicar(5,5)<<std::endl;
std::cout << "\n2do" << multiplicar(5,5,5)<<std::endl;
std::cout << "\nJoaquin Marcos Maita Flores 2025/10/06 hrs. 21:27--todos los
derechos reservados " <<std::endl;

// Ejemplo de llamada ambigua (si no tuviéramos una versión exacta)
// Si solo tuviéramos sumar(double, double) y llamáramos sumar(5, 3),
// los 'int' se promocionarían a 'double'. ¡Pero aquí tenemos una exacta!
// std::cout << "Llamada con promocion (si no hubiera int,int): " << sumar(5,
(int)3.0) << std::endl;
// El casteo (int)3.0 no es necesario aquí, solo es para ilustrar.

return 0;
}
```

## CÓDIGO EJECUTADO

```
input
Suma de enteros (5, 3): Ejecutando sumar(int, int)... 8
Suma de doubles (5.5, 3.3): Ejecutando sumar(double, double)... 8.8
Concatenacion de strings ("Hola, ", "Mundo!"): Ejecutando sumar(const
std::string&, const std::string&)... Hola, Mundo!
Suma de tres enteros (1, 2, 3): Ejecutando sumar(int, int, int)... 6

ejecutando no una suma si no una multiplicacion por que es lo unico
que se me ocurrio ojala este bien el ejercicio

1ro
1ro = 25

2do
2do = 125

Joaquin Marcos Maita Flores 2025/10/06 hrs. 21:27--todos los derecho
s reservados

...Program finished with exit code 0
Press ENTER to exit console.
```

**Este ejemplo enseña:**

**1. Sobrecarga de funciones:**

- Mismo nombre de función, pero con diferentes tipos o cantidad de parámetros.
- El compilador decide cuál función ejecutar según los argumentos.

**2. Ejemplos de sobrecarga:**

- `sumar(int, int) → Suma enteros.`
- `sumar(double, double) → Suma doubles.`
- `sumar(string, string) → Concatena strings.`
- `sumar(int, int, int) → Suma tres enteros.`

**3. Ventajas:**

- **Legibilidad:** Usar el mismo nombre para operaciones similares (ej: sumar).
- **Flexibilidad:** Acepta diferentes tipos de datos sin cambiar el nombre de la función.