



Ejercicio 9

Docente: Jimmy Nataniel Requena Llorentty

Materia: Programación III

Carrera: Ingeniería En Sistemas

Estudiantes: Joaquin Marcos Maita Flores

Santa Cruz – Bolivia

2025

Plantillas: template

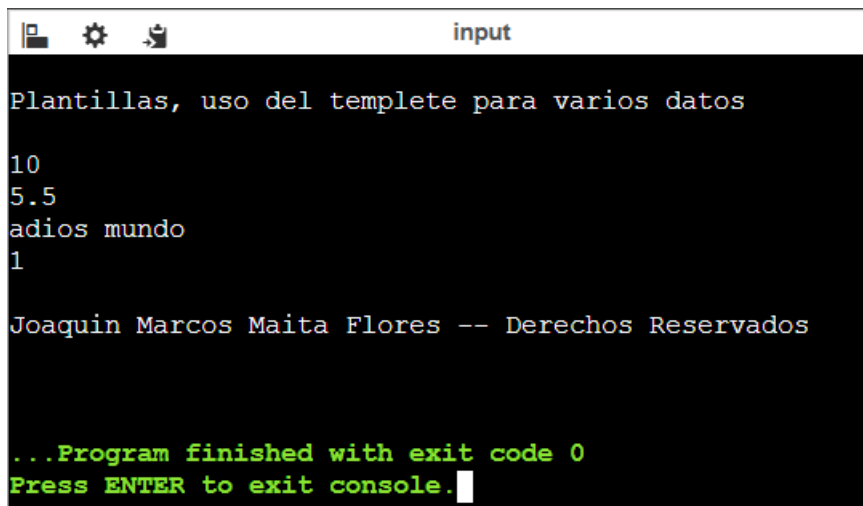
```
#include <iostream>
using namespace std;

template <typename T> // Plantilla genérica (T puede ser int, double, string,
etc.)
void imprimir(T a) { // Función que acepta cualquier tipo de dato
    cout << a << endl; // Imprime el valor, sin importar si es un número, texto o
booleano.
}

int main() {
    std::cout<<" \nPlantillas, uso del template para varios datos\n"<<std::endl;
    imprimir(10);           // int o enteros
    imprimir(5.5);          // double o decimales
    imprimir("adios mundo"); // string o caracteres
    imprimir(true);         // bool (se imprime como 1 o "true" según el
compilador)
    std::cout<<" \nJoaquin Marcos Maita Flores -- Derechos
Reservados\n"<<std::endl;

    return 0;
}
```

CÓDIGO EJECUTADO



```
input

Plantillas, uso del template para varios datos

10
5.5
adios mundo
1

Joaquin Marcos Maita Flores -- Derechos Reservados

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

¿Qué es un template?

Un **template (plantilla)** en C++ es una herramienta que permite escribir código genérico, es decir, funciones o clases que pueden trabajar con **diferentes tipos de datos** sin tener que reescribir el código para cada tipo.

¿Por qué usar templates?

- **Evita repetir código:** No necesitas escribir una función/clase para cada tipo (int, float, string, etc.).
- **Mantenimiento más fácil:** Si hay un error, solo lo corriges en el template.
- **Flexibilidad:** Funciona con cualquier tipo de dato (incluso clases personalizadas).