



### **Actividad 16**

Docente: Jimmy Nataniel Requena Llorentty

Materia: Programación III

Carrera: Ingeniería En Sistemas

Estudiantes: Joaquin Marcos Maita Flores

Santa Cruz – Bolivia

2025

## ¡Diseñemos Juntos! Invertir una Cadena

Problema: Escribir una función `std::string invertir(std::string s)` que devuelva la cadena `s` invertida.

- Ej: `invertir("hola")` debe devolver `"aloh"`.

```
#include <iostream>
#include <string>
#include <algorithm> // Para std::reverse (solo para comparar)

std::string invertirRecursiva(const std::string& s) {
    // std::cout << "Llamada con: \"" << s << "\"" << std::endl; // Para traza

    // Caso Base: Cadena vacía o de un solo carácter
    if (s.length() <= 1) {
        // std::cout << " Caso Base, retorna: \"" << s << "\"" << std::endl;
        return s;
    }
    // Paso Recursivo:
    else {
        char primerCaracter = s[0];
        std::string restoDeLaCadena = s.substr(1);
        // std::cout << " Primer caracter: '" << primerCaracter
        //          << "', Resto: \"" << restoDeLaCadena << "\"" << std::endl;

        std::string restoInvertido = invertirRecursiva(restoDeLaCadena); // ¡Fe
recursiva!
        // std::cout << " Resto invertido: \"" << restoInvertido << "\"" <<
std::endl;

        return restoInvertido + primerCaracter; // Combinar
    }
}

int main() {
    std::string texto;
    std::cout << "ingresa solo una palabra"<<std::endl;
    std::cin >> texto ;
```

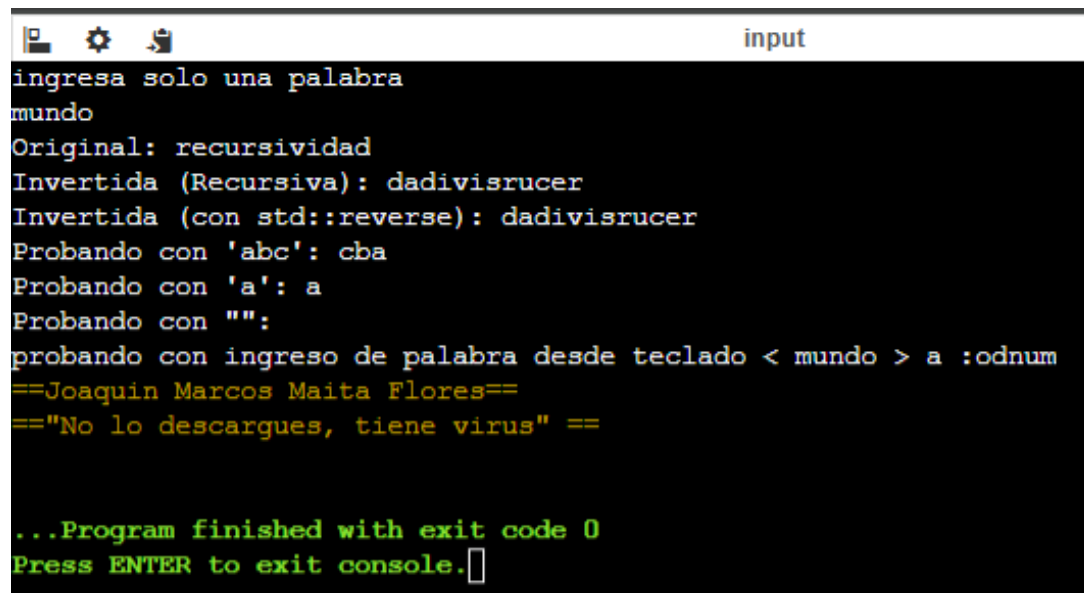
```
std::string original = "recursividad";
std::string invertida = invertirRecursiva(original);

std::cout << "Original: " << original << std::endl;
std::cout << "Invertida (Recursiva): " << invertida << std::endl;

// Para comparar (no es parte de la solución recursiva)
std::string comparacion = original;
std::reverse(comparacion.begin(), comparacion.end());
std::cout << "Invertida (con std::reverse): " << comparacion << std::endl;

std::cout << "Probando con 'abc': " << invertirRecursiva("abc") << std::endl;
std::cout << "Probando con 'a': " << invertirRecursiva("a") << std::endl;
std::cout << "Probando con \"\": " << invertirRecursiva("") << std::endl;
std::cout << "probando con ingreso de palabra desde teclado < " + texto + " >
a : " << invertirRecursiva (texto)<<std::endl;
std::cout << "\033[33m==Joaquin Marcos Maita Flores==\033[0m" << std::endl;
std::cout << "\033[33m==\"No lo descargues, tiene virus\" ==\033[0m" <<
std::endl;
return 0;
}
```

Código Corriendo



```
input
ingresa solo una palabra
mundo
Original: recursividad
Invertida (Recursiva): dadivisrucer
Invertida (con std::reverse): dadivisrucer
Probando con 'abc': cba
Probando con 'a': a
Probando con "":
probando con ingreso de palabra desde teclado < mundo > a :odnum
==Joaquin Marcos Maita Flores==
=="No lo descargues, tiene virus" ==

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

**¿Que nos enseña este código?**

**1. Recursión pura**

- Divide el problema en versiones más pequeñas de sí mismo
- Cada llamada resuelve parte del problema y delega el resto

**2. Manejo de strings**

- Uso de substr() para dividir cadenas
- Concatenación con + para construir resultados

**3. Casos base**

- Condición de salida (s.size() <= 1)
- Evita recursión infinita

**4. Modelo LIFO (Último en Entrar, Primero en Salir)**

- El último carácter procesado es el primero en el resultado final

**Anexo**

<https://onlinegdb.com/WaQ97CxnL>