

**AVANCE 1**

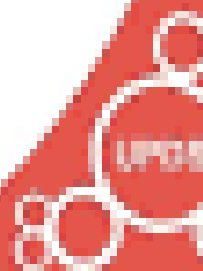
## **PROGRAMACION 3**

**ALUMNO: JHONNY PEREZ BELLIDO**

**DOCENTE: JIMMY NATNIEL REQUENA LL.**

**TURNO: NOCHE**

**JUNIO-2025**



# SUMA PROGRES

```
principal.cpp principal.cpp
1. /*****
2
3 Bienvenido a GDB Online.
4 GDB en línea es una herramienta de compilador y depuradora en línea para C, C++,
5 C, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, HTML, CSS, JS
6 Código, Compile, Run y Depura en línea desde cualquier parte del mundo.
7
8 *****/
9 #include <iostream> // Parador::cout, std::endl
10 Include .string // Para std::string
11
12 // Versión 1: Suma dos enteros
13 int sumar(int a, int b){
14     " std::cout<<"Ejecutando sumar (int, int)... " ;
15     devolver a +b .
16 }
17
18 // Versión 2: Suma dos números de punto flotante (doble)
```

entrada de entrada

```
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

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

Activar Windows  
Ve a Configuración para activar Windows.

# MULTIPLE

```
main.cpp
15 }
16
17 // Área de un rectángulo
18 double calcularArea(double base, double altura) {
19     std::cout << "Calculando área de RECTÁNGULO..." << std::endl;
20     return base * altura;
21 }
22
23 // Área de un cuadrado
24 double calcularArea(int lado) {
25     std::cout << "Calculando área de CUADRADO..." << std::endl;
26     return lado * lado;
27 }
28
29 int main() {
30     double areaCirc = calcularArea(5.0); // Llama a la función de círculo
31     double areaRect = calcularArea(4.0, 6.0); // Llama a la función de rectángulo
32     double areaCuad = calcularArea(7); // Llama a la función de cuadrado
33
34     std::cout << "Área del círculo: " << areaCirc << std::endl;
35     std::cout << "Área del rectángulo: " << areaRect << std::endl;
36     std::cout << "Área del cuadrado: " << areaCuad << std::endl;
37     return 0;
38 }
39
```

input

```
rea del cuadrado: 49


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

Activar Windows  
Ve a Configuración para activar Windows.

Lupo Vicepresidente  
Unidad

**ALIANZA  
UNIDAD**

EN 100 DÍAS  
VAMOS A CAMBIAR  
LAS EXPECTATIVAS,  
Abrir un FONDO DE



# MOSTRAR COMPLETO

```
main.cpp
15 }
16
17 // Área de un rectángulo
18 double calcularArea(double base, double altura) {
19     std::cout << "Calculando área de RECTÁNGULO..." << std::endl;
20     return base * altura;
21 }
22
23 // Área de un cuadrado
24 double calcularArea(int lado) {
25     std::cout << "Calculando área de CUADRADO..." << std::endl;
26     return lado * lado;
27 }
28
29 int main() {
30     double areaCirc = calcularArea(5.0); // Llama a la función de círculo
31     double areaRect = calcularArea(4.0, 6.0); // Llama a la función de rectángulo
32     double areaCua = calcularArea(7); // Llama a la función de cuadrado
33
34     std::cout << "Área del círculo: " << areaCirc << std::endl;
35     std::cout << "Área del rectángulo: " << areaRect << std::endl;
36     std::cout << "Área del cuadrado: " << areaCua << std::endl;
37     return 0;
38 }
39
```

Área del cuadrado: 49

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

# MOSTRAR NOMBRE

```
main.cpp
39     std::cout << "Tipo Entero (int): " << valor << std::endl;
40 }
41
42 void mostrar(double valor) {
43     std::cout << "Tipo Decimal (double): " << valor << std::endl;
44 }
45
46 void mostrar(const std::string& valor) {
47     std::cout << "Tipo Cadena (std::string): \"" << valor << "\"" << std::endl;
48 }
49
50 void mostrar(char valor) {
51     std::cout << "Tipo Caracter (char): '" << valor << "'" << std::endl;
52 }
53
54 void mostrar(const std::vector<int>& miVector) {
55     std::cout << "Tipo Vector de Enteros (std::vector<int>): [ ";
56     for (size_t i = 0; i < miVector.size(); ++i) {
57         std::cout << miVector[i] << (i == miVector.size() - 1 ? " " : ", ");
58     }
59     std::cout << " ]" << std::endl;
60     mostrar("JHONNY");
61 }
```

Tipo Vector de Enteros (std::vector<int>): [ 10, 20, 30, 40, 50 ]  
Tipo Cadena (std::string): "JHONNY"  
Tipo Cadena (std::string): "Esto es un literal de C-string."

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



# VALIDAR ENTERO

```
main.cpp
34 int num = 50;
35 cout << "¿El int " << num << " es válido? " << (validarEntrada(num) ? "Sí" : "No");
36
37 double dec = 5.5;
38 cout << "¿El double " << dec << " es válido? " << (validarEntrada(dec) ? "Sí" : "No");
39
40 string texto = "Hola";
41 cout << "¿El string \" " << texto << "\" es válido? " << (validarEntrada(texto) ? "Sí" : "No");
42
43 cout << "\n-- Pruebas con datos inválidos --\n";
44 cout << "¿El int -5 es válido? " << (validarEntrada(-5) ? "Sí" : "No") << endl;
45 cout << "¿El double 12.3 es válido? " << (validarEntrada(12.3) ? "Sí" : "No") << endl;
46 cout << "¿El string \"H0l@\" es válido? " << (validarEntrada("H0l@") ? "Sí" : "No");
47
48 mostrar("JHONNY"); // + ahora sí funcionará
49
50 return 0;
51
52
53 void mostrar(const string& nombre) {
54     cout << "Hola, " << nombre << "!" << endl;
55 }
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

El double 12.3 es válido? No  
El string "H0l@" es válido? No  
ola, JHONNY!

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

# FACTORES

```
main.cpp
12 // Caso Base
13 if (n == 0 || n == 1) {
14     std::cout << " factorial(" << n << ") -> Caso Base! Retorna 1." << std::endl;
15     return 1;
16 }
17 // Paso Recursivo
18 else {
19     std::cout << " factorial(" << n << ") -> Paso Recursivo. Llama a factorial(" << n - 1 << ")";
20     long long resultadoRecursion = factorial(n - 1); // ¡Auto-llamada!
21     long long resultadoFinal = n * resultadoRecursion;
22     std::cout << " factorial(" << n << ") -> Retornando " << n << " * " << resultadoRecursion << " = " << resultadoFinal << std::endl;
23     return resultadoFinal;
24 }
25
26
27 int main() {
28     int numero = 4; // Probemos con 4!
29     std::cout << "Iniciando calculo del factorial de " << numero << "." << std::endl;
30     long long resultado = factorial(numero);
31     std::cout << "\nEl factorial de " << numero << " es: " << resultado << std::endl;
32     return 0;
33 }
34
```

factorial(4) -> Retornando 4 \* 6 = 24

1 factorial de 4 es: 24

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

# VALIDAR ENTERO 1

```
main.cpp
31 }
32
33 int main() {
34     // Probar con int
35     int num = 50;
36     cout << "¿El int " << num << " es válido? " << (validarEntrada(num) ? "Sí" : "No") << endl;
37
38     // Probar con double
39     double dec = 5.5;
40     cout << "¿El double " << dec << " es válido? " << (validarEntrada(dec) ? "Sí" : "No") << endl;
41
42     // Probar con string
43     string texto = "Hola";
44     cout << "¿El string \"" << texto << "\" es válido? " << (validarEntrada(texto) ? "Sí" : "No") << endl;
45
46     // Probar con valores inválidos
47     cout << "\n-- Pruebas con datos inválidos --\n";
48     cout << "¿El int -5 es válido? " << (validarEntrada(-5) ? "Sí" : "No") << endl;
49     cout << "¿El double 12.3 es válido? " << (validarEntrada(12.3) ? "Sí" : "No") << endl;
50     cout << "¿El string \"H0l@\" es válido? " << (validarEntrada("H0l@") ? "Sí" : "No") << endl;
51
52     return 0;
53 }
```

input

```
¿El int -5 es válido? No
¿El double 12.3 es válido? No
¿El string "H0l@" es válido? No

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

Activar Windows  
Ve a Configuración para activar Windows.

# CONSTRUYENDO INVERSION

```
main.cpp
22 std::string restoDeLaCadena = s.substr(1);
23 // std::cout << " Primer caracter: " << primerCaracter
24 // << ", Resto: \"" << restoDeLaCadena << "\" << std::endl;
25 std::string restoInvertido = invertirRecursiva(restoDeLaCadena); // ¡Fe recursiva!
26 // std::cout << " Resto invertido: \"" << restoInvertido << "\" << std::endl;
27 return restoInvertido + primerCaracter; // Combinar
28 }
29 }
30 int main() {
31     std::string original = "recursividad";
32     std::string invertida = invertirRecursiva(original);
33     std::cout << "Original: " << original << std::endl;
34     std::cout << "Invertida (Recursiva): " << invertida << std::endl;
35     // Para comparar (no es parte de la solución recursiva)
36     std::string comparacion = original;
37     std::reverse(comparacion.begin(), comparacion.end());
38     std::cout << "Invertida (con std::reverse): " << comparacion << std::endl;
39     std::cout << "Probando con 'abc': " << invertirRecursiva("abc") << std::endl;
40     std::cout << "Probando con 'a': " << invertirRecursiva("a") << std::endl;
41     std::cout << "Probando con \"\": " << invertirRecursiva("") << std::endl;
42     return 0;
43 }
44 }
```

input

```
Probando con 'abc': cba
Probando con 'a': a
Probando con "":

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

Activar Windows  
Ve a Configuración para activar Windows.



