

UNIVERSIDAD PRIVADA DOMINGO SAVIO



LICENCIATURA EN INGENIERÍA EN SISTEMAS

"EJERCICIOS AVANZADOS"

GRUPO

CUARTETO INDOMABLE

Jhon Emanuel Flores Chambi

Bernardo Ribera B

Joaquín Marcos Maita Flor

DOCENTE

Jimmy Nataniel Requena Llorentty

Santa Cruz de la Sierra – Bolivia

2025

Ejercicio – Pizzas

```
Bro
21
pizza1

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

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

Ejercicio - Punteros en acción

```
--- Información de 'variable' ---
Valor de 'variable': 20
Dirección de 'variable' (&variable): 0x7ffc07f1bf0c

--- Información de 'puntero' ---
Contenido de 'puntero' (la dirección que guarda): 0x7ffc07f1bf0c
Dirección donde está guardado el propio 'puntero' (&puntero): 0x7ffc07f1bf10

--- Accediendo al valor A TRAVÉS del puntero ---
Valor al que apunta 'puntero' (*puntero): 20

--- Modificando a través del puntero ---
Nuevo valor de 'variable' (después de *puntero = 30): 30
Nuevo valor apuntado por 'puntero' (*puntero): 30

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi
```

Ejercicio - Gestión dinámica

```
Entero dinámico creado. Valor: 123 en dirección: 0x61a16f5582b0
Memoria del entero dinámico liberada.

--- Arreglo Dinámico ---
Arreglo dinámico creado y llenado:
p_arreglo_doubles[0] = 0 en dir: 0x61a16f5586e0
p_arreglo_doubles[1] = 1.5 en dir: 0x61a16f5586e8
p_arreglo_doubles[2] = 3 en dir: 0x61a16f5586f0
p_arreglo_doubles[3] = 4.5 en dir: 0x61a16f5586f8
p_arreglo_doubles[4] = 6 en dir: 0x61a16f558700
Memoria del arreglo dinámico liberada.

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi
```

Ejercicio –

Minicadena

```
Creado primer nodo (cabeza) con dato: 10
Creado segundo nodo con dato: 20
Enlazando cabeza->siguiente con segundoNodo.
Creado tercer nodo con dato: 30
Enlazando segundoNodo->siguiente con tercerNodo.

Recorriendo la mini-lista:
Dato en cabeza: 10
Dato en el segundo nodo (via cabeza->siguiente): 20
Dato en el tercer nodo (via cabeza->siguiente->siguiente): 30

Liberando memoria...
Tercer nodo liberado.
Segundo nodo liberado.
Primer nodo (cabeza) liberado.

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi
```

Ejercicio – Sobrecarga de sumar

```
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

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi
```

Ejercicio – Mostrar completo 1

```
Entero: 100
Double: 3.14159
String: "Hola POO!"
Caracter: 'A'
Vector de enteros: [1, 2, 3, 4, 5]

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi
```

Ejercicio –

Nombre multiples

```
Calculando área de CÍRCULO...
Calculando área de RECTÁNGULO...
Calculando área de CUADRADO...
Área del círculo: 78.5397
Área del rectángulo: 24
Área del cuadrado: 49

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

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

Ejercicio – Mostrar punto

```
=== Demostrando constructores ===
Punto creado en el origen (0,0) por constructor por defecto.
Punto creado en (5,3) por constructor con coords.
Punto copiado de (5,3).

=== Mostrando puntos ===
Punto(0, 0)
Punto(5, 3)
Punto(5, 3)

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi
Punto en (5,3) destruido.
Punto en (5,3) destruido.
Punto en (0,0) destruido.
```

Ejercicio – Mostrar Completo 2

```
--- Demostracion de 'mostrar' sobrecargado ---  
Tipo Entero (int): 100  
Tipo Decimal (double): 3.14159  
Tipo Cadena (std::string): "Hola desde Programacion III!"  
Tipo Caracter (char): 'Z'  
Tipo Vector de Enteros (std::vector<int>): [ 10, 20, 30, 40, 50 ]  
Tipo Cadena (std::string): "Esto es un literal de C-string."  
  
CUARTETO INDOMABLE  
Bernardo Ribera B  
Joaquín Marcos Maita Flor  
Jhon Emanuel Flores Chambi
```

Ejercicio – Coleccion poliformica

```
Dibujar Círculo  
Dibujar Cuadrado  
Dibujar Triángulo  
  
CUARTETO INDOMABLE  
Bernardo Ribera B  
Joaquín Marcos Maita Flor  
Jhon Emanuel Flores Chambi  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Ejercicio – Ejercicio – figura

```
Dibujar Círculo
Dibujar Cuadrado

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

...Program finished with exit code 0
```


unique_ptr

```
Dibujar Círculo
Dibujar Cuadrado
Dibujar Triángulo

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

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

Ejercicio – Ejercicio – Validar entrada



```
¿El int 50 es válido? Sí
¿El double 5.5 es válido? Sí
¿El string "Hola" es válido? Sí

-- Pruebas con datos inválidos --
¿El int -5 es válido? No
¿El double 12.3 es válido? No
¿El string "H0l@" es válido? No

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

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


Ejercicio – factorial

```
Inicio calculo del factorial de 4.  
Calculando factorial(4)...  
factorial(4) -> Paso Recursivo. Llama a factorial(3).  
Calculando factorial(3)...  
factorial(3) -> Paso Recursivo. Llama a factorial(2).  
Calculando factorial(2)...  
factorial(2) -> Paso Recursivo. Llama a factorial(1).  
Calculando factorial(1)...  
factorial(1) -> Caso Base! Retorna 1.  
factorial(2) -> Retornando 2 * 1 = 2  
factorial(3) -> Retornando 3 * 2 = 6  
factorial(4) -> Retornando 4 * 6 = 24
```

< El factorial de 4 es: 24

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

Ejercicio – Fibonacci

```
Secuencia de Fibonacci (primeros 7 terminos):  
0 1 1 2 3 5 8
```

CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

Ejercicio – Sumado de cadena

```
La suma recursiva del arreglo es: 100
```

< CUARTETO INDOMABLE
Bernardo Ribera B
Joaquín Marcos Maita Flor
Jhon Emanuel Flores Chambi

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



```
main.cpp
11
12 // Paso recursivo
13 return fibonacci(n - 1) + fibonacci(n - 2);
14 }
15
16 int main() {
17     const int terminos = 7; // Calcula hasta F(6); se imprimen 7 términos empezando en F(0)
18
19     std::cout << "Secuencia de Fibonacci (primeros "
20               << terminos << " términos):\n";
21
22     for (int i = 0; i < terminos; ++i) {
23         std::cout << fibonacci(i) << ' ';
24     }
25     std::cout << '\n';
26
27     // Descomenta para probar un término concreto
28     // std::cout << "F(7) es: " << fibonacci(7) << '\n';
29     std::cout << "BERNARDO RIBERA BARBERY..." << std::endl;
30
31     return 0;
32 }
```

input

Secuencia de Fibonacci (primeros 7 términos):
0 1 1 2 3 5 8
BERNARDO RIBERA BARBERY...

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


```
GDB x SC5F x [Prog x Issues x serie x Prese x Fibon x (15) V x G sprint x +
```

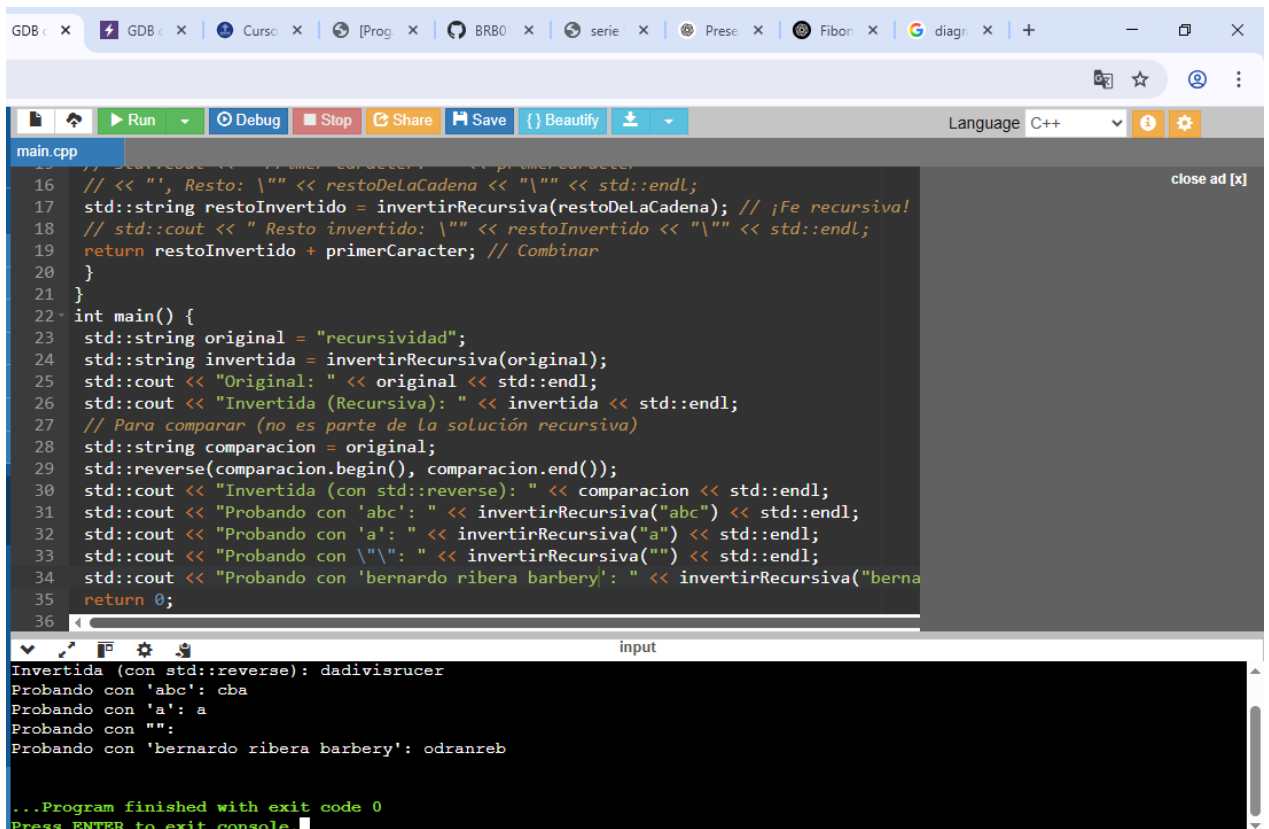
```
main.cpp
1 #include <iostream>
2 #include <vector>
3 // Suma los elementos de 'arr' desde el índice 'idx' hasta el final
4 int sumarArreglo(const std::vector<int>& arr, int idx) {
5     // Caso Base: Si el índice está fuera de los límites del vector,
6     // significa que no hay más elementos que sumar.
7     if (idx >= arr.size()) {
8         return 0;
9     }
10    // Paso Recursivo: Suma el elemento actual (arr[idx])
11    // con la suma del resto del arreglo (desde idx + 1).
12    else {
13        return arr[idx] + sumarArreglo(arr, idx + 1);
14    }
15 }
16 int main() {
17     std::vector<int> misNumeros = {1, 5, 5, 20, 5}; // Suma = 100
18     int sumaTotal = sumarArreglo(misNumeros, 0);
19     // Empezar desde el índice 0
20     std::cout << "La suma recursiva del arreglo es: " << sumaTotal << std::endl;
21     std::cout << "BERNARDO RIBERA BARBERY..." << std::endl;
22     return 0;
23 }
```

input

La suma recursiva del arreglo es: 36
BERNARDO RIBERA BARBERY...

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





```

16 // << " ", Resto: \"\" << restoDeLaCadena << \"\" << std::endl;
17 std::string restoInvertido = invertirRecursiva(restoDeLaCadena); // ¡Fe recursiva!
18 // std::cout << " Resto invertido: \"\" << restoInvertido << \"\" << std::endl;
19 return restoInvertido + primerCaracter; // Combinar
20 }
21 }
22 int main() {
23     std::string original = "recursividad";
24     std::string invertida = invertirRecursiva(original);
25     std::cout << "Original: " << original << std::endl;
26     std::cout << "Invertida (Recursiva): " << invertida << std::endl;
27     // Para comparar (no es parte de la solución recursiva)
28     std::string comparacion = original;
29     std::reverse(comparacion.begin(), comparacion.end());
30     std::cout << "Invertida (con std::reverse): " << comparacion << std::endl;
31     std::cout << "Probando con 'abc': " << invertirRecursiva("abc") << std::endl;
32     std::cout << "Probando con 'a': " << invertirRecursiva("a") << std::endl;
33     std::cout << "Probando con \"\": " << invertirRecursiva("") << std::endl;
34     std::cout << "Probando con 'bernardo ribera barbery': " << invertirRecursiva("berna
35     return 0;
36

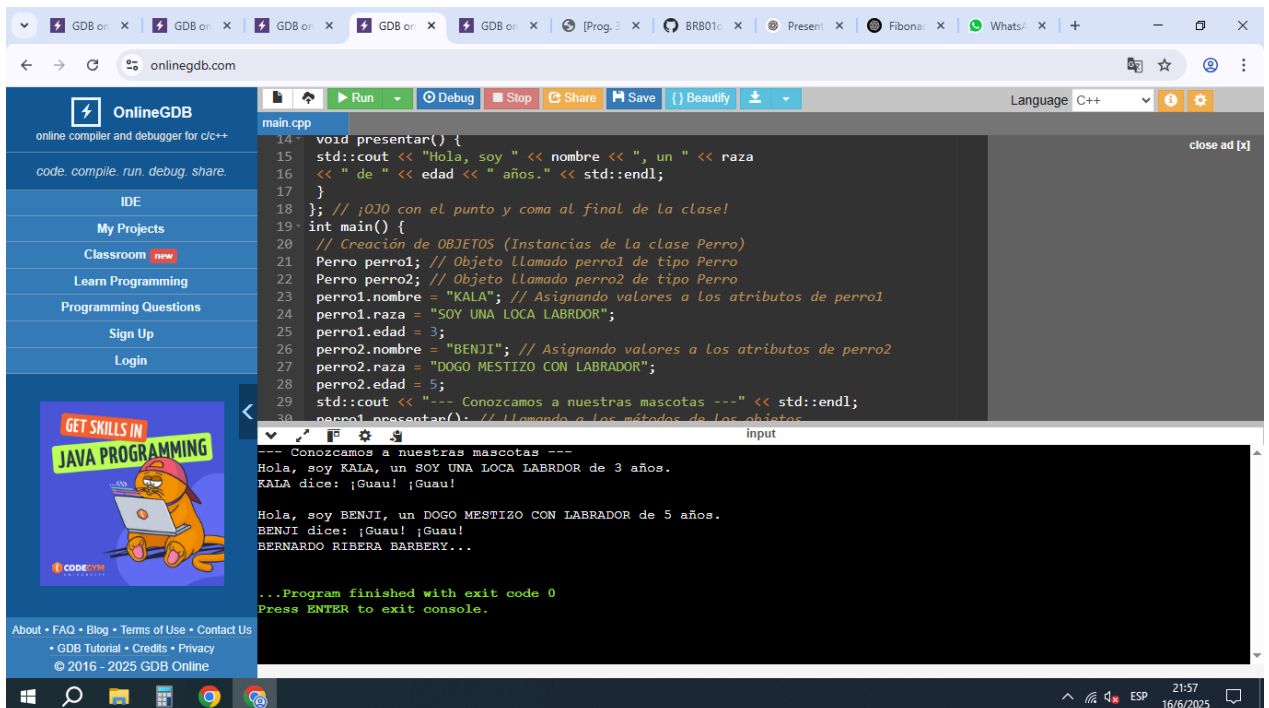
```

input

```

Invertida (con std::reverse): dadivisrucer
Probando con 'abc': cba
Probando con 'a': a
Probando con "":
Probando con 'bernardo ribera barbery': odranreb
...Program finished with exit code 0
Press ENTER to exit console.

```



```

14 void presentar() {
15     std::cout << "Hola, soy " << nombre << ", un " << raza
16     << " de " << edad << " años." << std::endl;
17 }
18 }; // ¡OJO con el punto y coma al final de la clase!
19 int main() {
20     // Creación de OBJETOS (Instancias de la clase Perro)
21     Perro perro1; // Objeto llamado perro1 de tipo Perro
22     Perro perro2; // Objeto llamado perro2 de tipo Perro
23     perro1.nombre = "KALA"; // Asignando valores a los atributos de perro1
24     perro1.raza = "SOY UNA LOCA LABRDOR";
25     perro1.edad = 3;
26     perro2.nombre = "BENJI"; // Asignando valores a los atributos de perro2
27     perro2.raza = "DOGO MESTIZO CON LABRADOR";
28     perro2.edad = 5;
29     std::cout << "--- Conozcamos a nuestras mascotas ---" << std::endl;
30     perro1.presentar(); // Llamando a los métodos de los objetos

```

input

```

--- Conozcamos a nuestras mascotas ---
Hola, soy KALA, un SOY UNA LOCA LABRDOR de 3 años.
KALA dice: ¡Guau! ¡Guau!

Hola, soy BENJI, un DOGO MESTIZO CON LABRADOR de 5 años.
BENJI dice: ¡Guau! ¡Guau!
BERNARDO RIBERA BARBERY...
...Program finished with exit code 0
Press ENTER to exit console.

```

```

input
Información actualizada de Joaquín Soliz:
-----
Nombre: Joaquín Soliz
Edad: 21 años
Matricula: A123
Promedio: 8.5
-----
Estudiante 'Priscila Vaca' creado.
-----
Nombre: Priscila Vaca
Edad: 19 años
Matricula: B456
Promedio: 0
-----
Intentando actualizar edad y promedio...
Error: Edad '119' invalida para el estudiante Priscila Vaca. Edad no modificada.
Error: Promedio '-6' invalido para Priscila Vaca. Promedio no modificado.

Información actualizada de Priscila Vaca:
-----
Nombre: Priscila Vaca
Edad: 19 años
Matricula: B456
Promedio: 9.2
-----
bernardo ribera barbery:
ENCAPSULAMIENTO A EL ESTUDIANTE:

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

```

```

91     std::cout << "\nIntentando actualizar edad, promedio y correo..." << std::endl;
92     estudiante1.setEdad(21);
93     estudiante1.setPromedio(8.5);
94     estudiante1.setCorreo("correo-invalido"); // Prueba de validación
95     estudiante1.setEdad(150);
96
input
stderr
close ad [x]

```

ompilation failed due to following error(s).

```

main.cpp: In function 'int main()':
main.cpp:102:18: error: 'int Estudiante::edad' is private within this context
102 |     estudiante1.edad = 25; // ERROR: 'edad' es privado
    |
    | ~~~~~
main.cpp:8:9: note: declared private here
8 |     int edad;
  |     ~~~~~
main.cpp:102:18: note: field 'int Estudiante::edad' can be accessed via 'int Estudiante::getEdad() const'
102 |     estudiante1.edad = 25; // ERROR: 'edad' es privado
    |
    | ~~~~~
    | getEdad()
main.cpp:103:31: error: 'double Estudiante::promedio' is private within this context
103 |     std::cout << estudiante1.promedio; // ERROR: 'promedio' también es privado
    |
    | ~~~~~
main.cpp:10:12: note: declared private here
10 |     double promedio;
   |     ~~~~~
main.cpp:103:31: note: field 'double Estudiante::promedio' can be accessed via 'double Estudiante::getPromedio() const'
103 |     std::cout << estudiante1.promedio; // ERROR: 'promedio' también es privado
    |
    | ~~~~~
    | getPromedio()

```

```
input
Intentando actualizar edad, promedio y correo...
Error: Correo electrónico inválido.
Error: Edad '150' inválida para el estudiante Juaquin Soliz.
Error: Promedio '-2' inválido para Juaquin Soliz.

Información actualizada de Juaquin Soliz:
-----
Nombre: Juaquin Soliz
Edad: 21 años
Matrícula: A123
Promedio: 8.5
Carrera: Ingeniería
Correo: juaquin@upds.edu
-----

Si descomentas las líneas anteriores y compilas, verás errores como:
'int Estudiante::edad' is private within this context
Esto enseña a respetar el encapsulamiento y a usar setters/getters.
Error: Edad '-10' inválida para el estudiante Priscila Vaca.
Error: Correo electrónico inválido.
Estudiante 'Priscila Vaca' creado.
-----
Nombre: Priscila Vaca
Edad: 558659616 años
Matrícula: B456
Promedio: 0
Carrera: No especificada
Correo:
-----
bernardo ribera barbery:
ENCAPSULAMIENTO A EL ESTUDIANTE:

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