

UNIVERSIDAD PRIVADA DOMINGO SAVIO



LICENCIATURA EN INGENIERÍA EN SISTEMAS

"EJERCICIOS AVANZADOS"

GRUPO

CUARTETO DINAMICO

Jhon Emanuel Flores Chambi.

Bernardo Ribera B.

Joaquín Marcos Maita Flor.

Rodrigo Muñoz Pizarro

Eduardo

DOCENTE

Jimmy Nataniel Requena Llorentty

Ejercicio –

Santa Cruz de la Sierra – Bolivia

2025

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 – Ejercicio - Gestión dinámica

```
Entero dinamico creado. Valor: 123 en direccion: 0x61a16f5582b0
Memoria del entero dinamico liberada.

--- Arreglo Dinamico ---
Arreglo dinamico 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 dinamico liberada.

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

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 – 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
```

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 – 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.
```

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 –

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 – 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

```
▼ ↗ 📄 ⚙️ 👤  
Iniciando 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) x Google x +
```

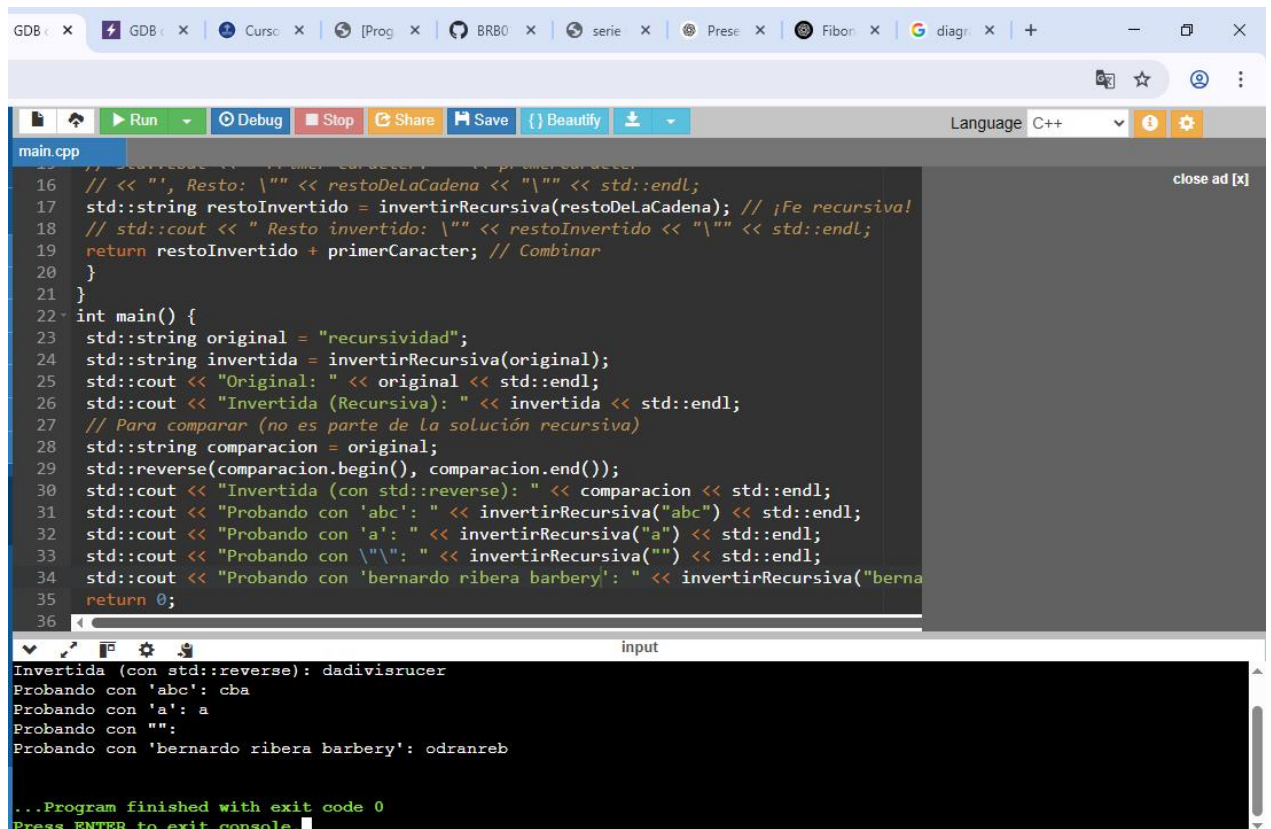
```
main.cpp
1 #include <iostream>
2 #include <vector>
3 // Suma los elementos de 'arr' desde el índice 'idx' hasta el final
4 int sumarArreglo(const std::vector<int>& arr, int idx) {
5     // Caso Base: Si el índice está fuera de los límites del vector,
6     // significa que no hay más elementos que 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

```

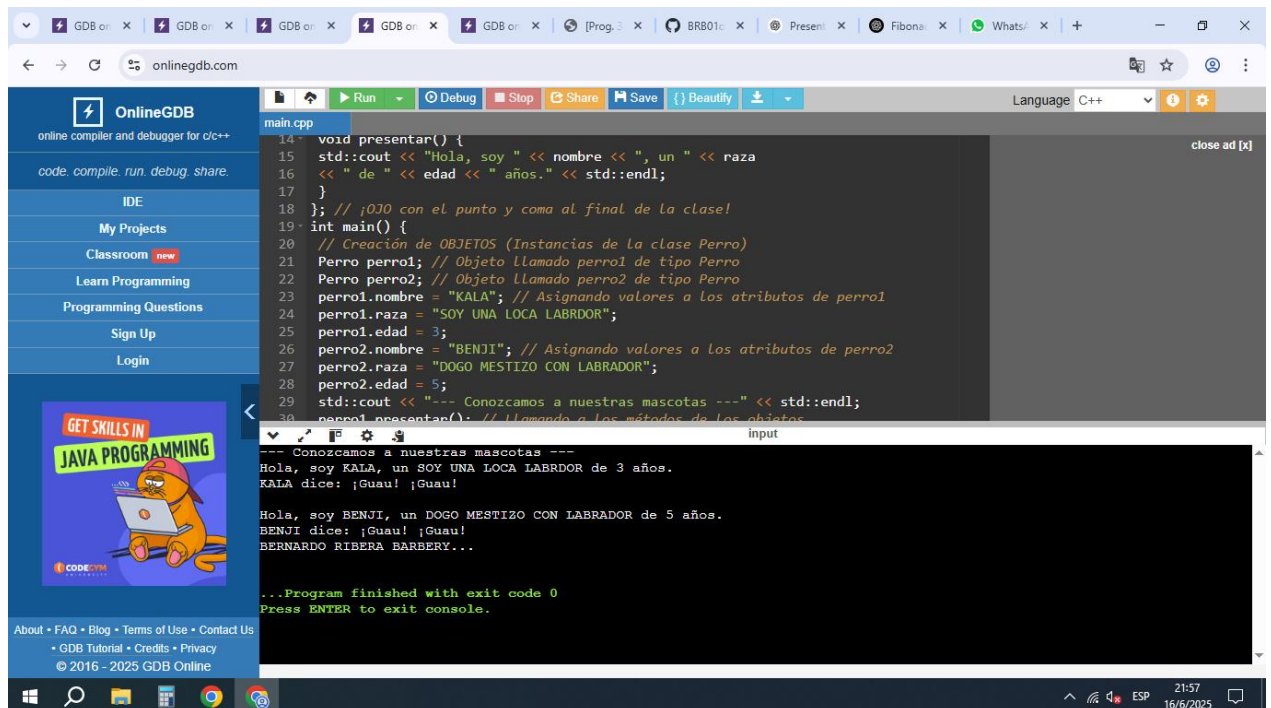
input

```

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
Informacion actualizada de Juaquin Soliz:
-----
Nombre: Juaquin 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.

Informacion 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

```

Compilation 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.
```

Código estudiante añadiendo mas atributos

```
input
E-mail: jsoliz@univ.edu

Actualizando datos válidos e inválidos...
Error: La carrera no puede estar vacía.
Error: Semestre '15' fuera de rango (1-12). No modificado.
Error: e-mail 'sin-arroba-ni-punto' con formato inválido. No modificado.

Información actualizada de Juaquín Soliz:
-----
Nombre: Juaquín Soliz
Edad: 21 años
Matrícula: A123
Promedio: 8.5
Carrera: Ing. de Sistemas
Semestre: 4
E-mail: jsoliz@univ.edu
-----

Creando estudiante2 con edad inválida (-10)...
Error: Edad '-10' inválida. Edad no modificada.
Estudiante 'Priscila Vaca' creado.
-----
Nombre: Priscila Vaca
Edad: 1549107184 años
Matrícula: B456
Promedio: 0
Carrera: Derecho
Semestre: 2
E-mail: pvaca@univ.edu
-----
BERNARDO RIBERA BARBERY

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



```
main.cpp
26 };
27 // Función principal
28 int main() {
29 // Crear un punto con el constructor por defecto
30 Punto p1;
31 p1.mostrar();
32 // Crear un punto con valores personalizados
33 Punto p2(5.5, 7.3);
34 p2.mostrar();
35 // Crear un punto copiando otro (constructor copia)
36 Punto p3(p2);
37 p3.mostrar();
38 std::cout << "BERNARDO RIBERA BARBERY " << std::endl;
39 return 0;
}

input
< Constructor por defecto llamado. x = 0, y = 0
Punto en coordenadas: (0, 0)
Constructor con parámetros llamado. x = 5.5, y = 7.3
Punto en coordenadas: (5.5, 7.3)
Constructor copia llamado. Copiando punto: x = 5.5, y = 7.3
Punto en coordenadas: (5.5, 7.3)
BERNARDO RIBERA BARBERY

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

Us
```

motor

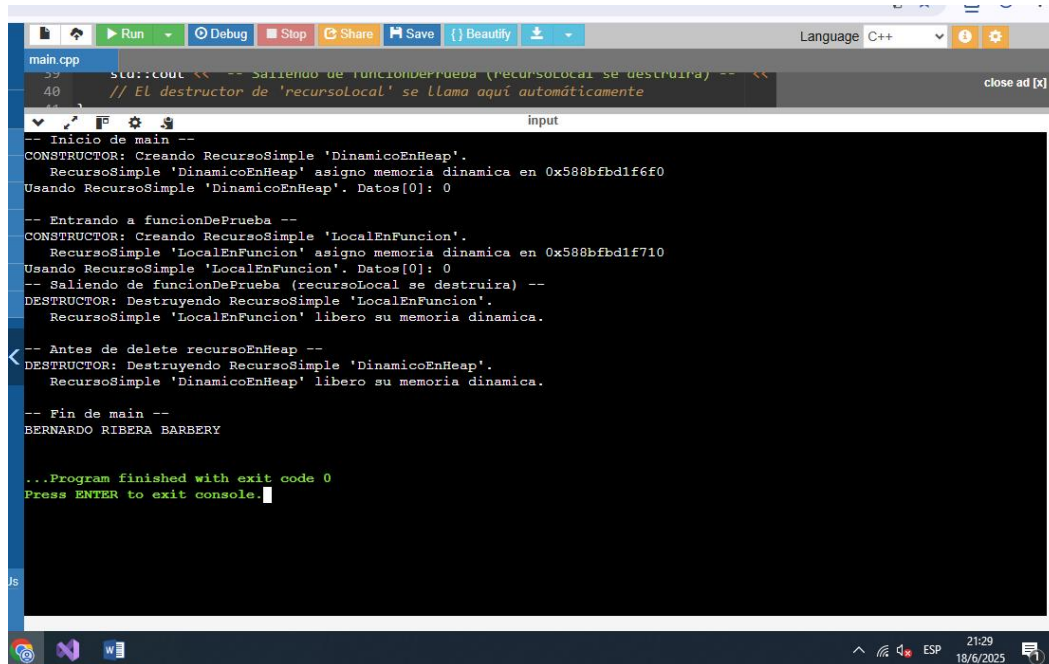
```
main.cpp
31 // Forma alternativa (menos eficiente)
32 Auto(std::string m, int cil) {
33     marca = m;
34     // Aquí primero se crea miMotor usando el constructor por defecto (Motor())
35     // Luego se reasigna con un nuevo Motor(cil), lo cual es menos eficiente:
36     // - Se hacen dos operaciones: construcción y luego reasignación.
37 }

input
Motor(int) creado con 6 cilindros
Auto 'Toyota' creado con motor de 6 cilindros
BERNARDO RIBERA BARBERY

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

Us
```

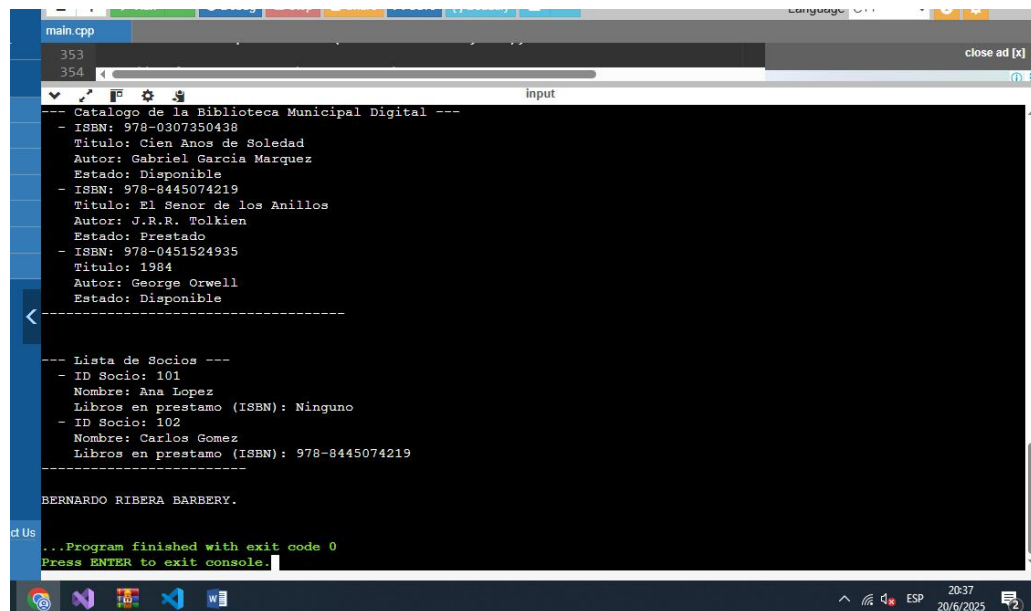
RECURSO SIMPLE



```
main.cpp
33 std::cout << "Saliendo de funcionDePrueba (recursoLocal se destruirá)" << "\n";
40 // EL destructor de 'recursoLocal' se llama aquí automáticamente
41
-- Inicio de main --
CONSTRUCTOR: Creando RecursoSimple 'DinamicoEnHeap'.
RecursoSimple 'DinamicoEnHeap' asigno memoria dinamica en 0x588bfb1f6f0
Usando RecursoSimple 'DinamicoEnHeap'. Datos[0]: 0
-- Entrando a funcionDePrueba --
CONSTRUCTOR: Creando RecursoSimple 'LocalEnFuncion'.
RecursoSimple 'LocalEnFuncion' asigno memoria dinamica en 0x588bfb1f710
Usando RecursoSimple 'LocalEnFuncion'. Datos[0]: 0
-- Saliendo de funcionDePrueba (recursoLocal se destruirá) --
DESTRUCTOR: Destruyendo RecursoSimple 'LocalEnFuncion'.
RecursoSimple 'LocalEnFuncion' libero su memoria dinamica.
-- Antes de delete recursoEnHeap --
DESTRUCTOR: Destruyendo RecursoSimple 'DinamicoEnHeap'.
RecursoSimple 'DinamicoEnHeap' libero su memoria dinamica.
-- Fin de main --
BERNARDO RIBERA BARBERY

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

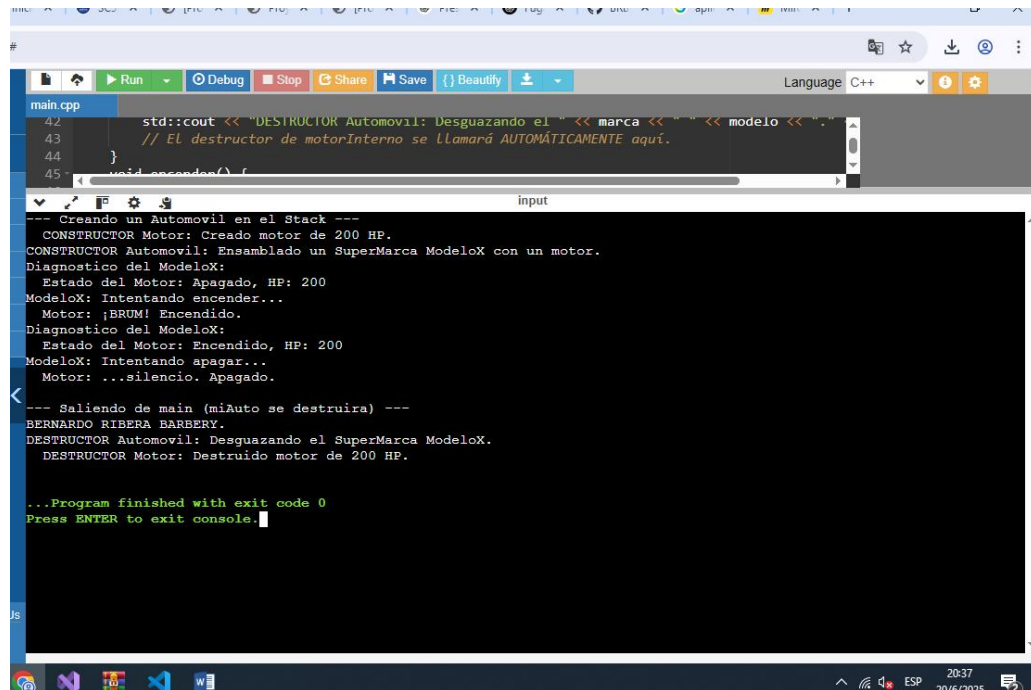
MINI SISTEMA



```
main.cpp
353
354
--- Catalogo de la Biblioteca Municipal Digital ---
- ISBN: 978-0307350438
  Titulo: Cien Anos de Soledad
  Autor: Gabriel Garcia Marquez
  Estado: Disponible
- ISBN: 978-8445074219
  Titulo: El Senor de los Anillos
  Autor: J.R.R. Tolkien
  Estado: Prestado
- ISBN: 978-0451524935
  Titulo: 1984
  Autor: George Orwell
  Estado: Disponible
-----
--- Lista de Socios ---
- ID Socio: 101
  Nombre: Ana Lopez
  Libros en prestamo (ISBN): Ninguno
- ID Socio: 102
  Nombre: Carlos Gomez
  Libros en prestamo (ISBN): 978-8445074219
-----
BERNARDO RIBERA BARBERY.

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


AUTOMOVIL



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

// Clase Automovil
class Automovil {
public:
    string marca;
    string modelo;
    int hp;
    bool encendido;

    Automovil(string m, string mo, int h) {
        marca = m;
        modelo = mo;
        hp = h;
        encendido = false;
    }

    void encender() {
        encendido = true;
    }

    void apagar() {
        encendido = false;
    }

    ~Automovil() {
        cout << "DESTRUCTOR Automovil: Desguazando el " << marca << " " << modelo << ".\n";
        // El destructor de motorInterno se llamará AUTOMÁTICAMENTE aquí.
    }
};

// Clase Motor
class Motor {
public:
    int hp;

    Motor(int h) {
        hp = h;
    }

    ~Motor() {
        cout << "DESTRUCTOR Motor: Destruído motor de " << hp << " HP.\n";
    }
};

// Clase SuperMarca
class SuperMarca {
public:
    string modeloX;

    SuperMarca(string m) {
        modeloX = m;
    }
};

int main() {
    cout << "--- Creando un Automovil en el Stack ---\n";
    // Constructor Motor: Creado motor de 200 HP.
    Motor motor(200);
    // Constructor Automovil: Ensamblado un SuperMarca ModeloX con un motor.
    Automovil miAuto("SuperMarca", "ModeloX", 200);

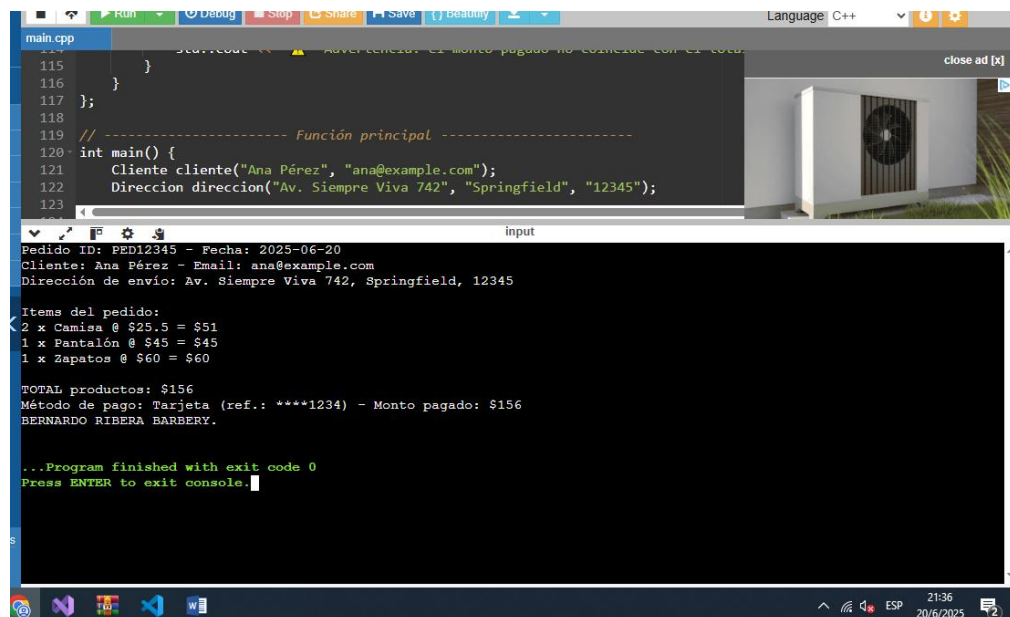
    // Diagnostico del ModeloX:
    cout << "Estado del Motor: Apagado, HP: 200\n";
    miAuto.encender();
    cout << "ModeloX: Intentando encender...\n";
    cout << "Motor: ¡BRUM! Encendido.\n";

    // Diagnostico del ModeloX:
    cout << "Estado del Motor: Encendido, HP: 200\n";
    miAuto.apagar();
    cout << "ModeloX: Intentando apagar...\n";
    cout << "Motor: ...silencio. Apagado.\n";

    cout << "--- Saliendo de main (miAuto se destruya) ---\n";
    // Destructor Automovil: Desguazando el SuperMarca ModeloX.
    // Destructor Motor: Destruído motor de 200 HP.

    return 0;
}
```

PEDIDO ONLINE



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

// Clase Cliente
class Cliente {
public:
    string nombre;
    string email;

    Cliente(string n, string e) {
        nombre = n;
        email = e;
    }
};

// Clase Pedido
class Pedido {
public:
    string id;
    string fecha;
    string cliente;
    string direccion;
    int items;
    float total;
    string metodo;

    Pedido(string i, string f, string c, string d, int it, float t, string m) {
        id = i;
        fecha = f;
        cliente = c;
        direccion = d;
        items = it;
        total = t;
        metodo = m;
    }
};

// Función principal
int main() {
    Cliente cliente("Ana Pérez", "ana@example.com");
    Pedido pedido("PED12345", "2025-06-20", cliente, "Av. Siempre Viva 742", "Springfield", "12345", "Tarjeta");

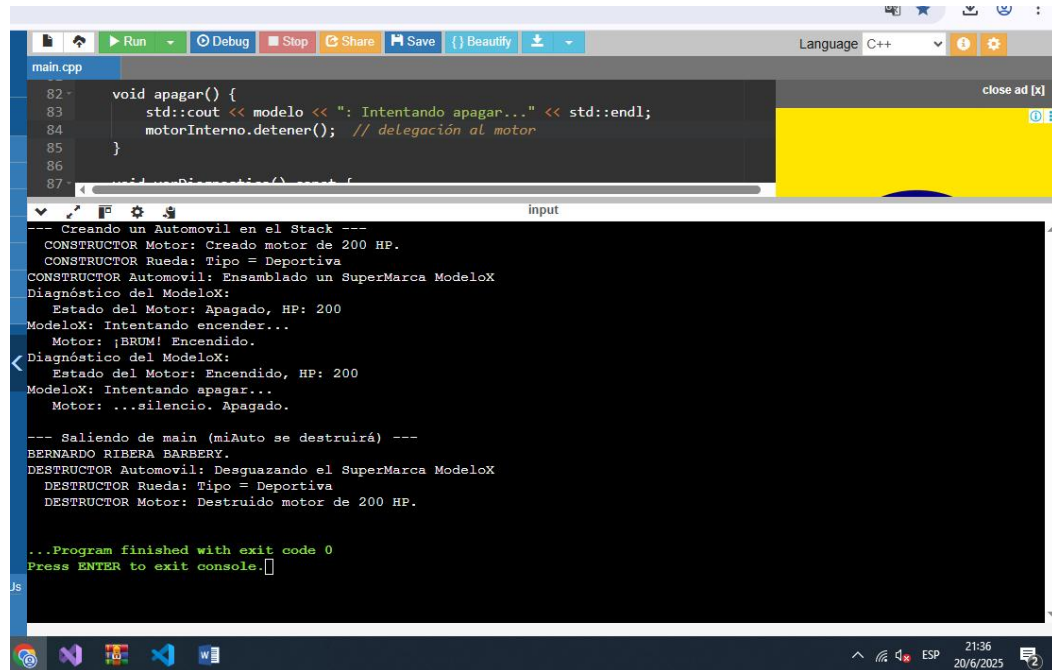
    cout << "Pedido ID: PED12345 - Fecha: 2025-06-20\n";
    cout << "Cliente: Ana Pérez - Email: ana@example.com\n";
    cout << "Dirección de envío: Av. Siempre Viva 742, Springfield, 12345\n";

    cout << "Items del pedido:\n";
    cout << "2 x Camisa @ $25.5 = $51\n";
    cout << "1 x Pantalón @ $45 = $45\n";
    cout << "1 x Zapatos @ $60 = $60\n";

    cout << "TOTAL productos: $156\n";
    cout << "Método de pago: Tarjeta (ref.: ****1234) - Monto pagado: $156\n";
    cout << "BERNARDO RIBERA BARBERY.\n";

    return 0;
}
```

AUTOMOVIL CON PARTES



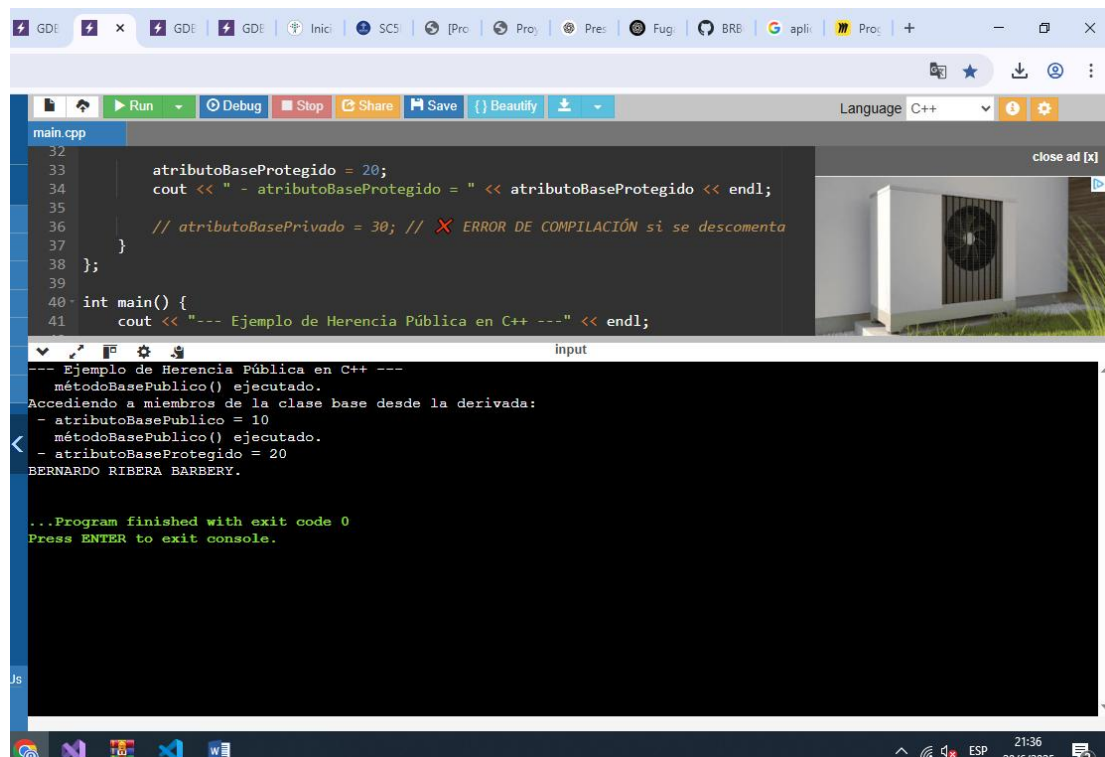
```

main.cpp
82 void apagar() {
83     std::cout << modelo << ": Intentando apagar..." << std::endl;
84     motorInterno.detener(); // delegación al motor
85 }
86
87
input
--- Creando un Automovil en el Stack ---
CONSTRUCTOR Motor: Creado motor de 200 HP.
CONSTRUCTOR Rueda: Tipo = Deportiva
CONSTRUCTOR Automovil: Ensamblado un SuperMarca ModeloX
Diagnóstico del ModeloX:
Estado del Motor: Apagado, HP: 200
ModeloX: Intentando encender...
Motor: ¡BRUM! Encendido.
< Diagnóstico del ModeloX:
Estado del Motor: Encendido, HP: 200
ModeloX: Intentando apagar...
Motor: ...silencio. Apagado.

--- Saliendo de main (miAuto se destruirá) ---
BERNARDO RIBERA BARBERY.
DESTRUCTOR Automovil: Desguazando el SuperMarca ModeloX
DESTRUCTOR Rueda: Tipo = Deportiva
DESTRUCTOR Motor: Destruído motor de 200 HP.

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

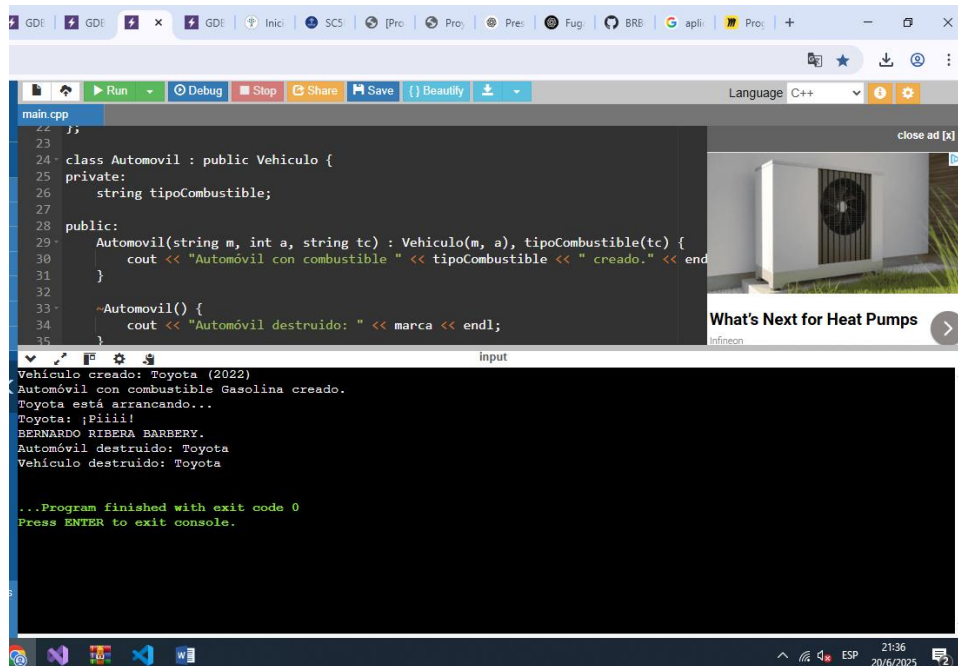
HERENCIA PUBLICA



```

main.cpp
32 atributoBaseProtegido = 20;
33 cout << " - atributoBaseProtegido = " << atributoBaseProtegido << endl;
34
35 // atributoBasePrivado = 30; // X ERROR DE COMPILACIÓN si se descomenta
36
37 }
38 };
39
40 int main() {
41     cout << "--- Ejemplo de Herencia Pública en C++ ---" << endl;
42
43     metodoBasePublico() ejecutado.
44     Accediendo a miembros de la clase base desde la derivada:
45     - atributoBasePublico = 10
46     metodoBasePublico() ejecutado.
47     - atributoBaseProtegido = 20
48     BERNARDO RIBERA BARBERY.
49
50 ...Program finished with exit code 0
51 Press ENTER to exit console.
  
```

AUTOMOVIL HERENCIA



```
main.cpp
22- }
23-
24- class Automovil : public Vehiculo {
25- private:
26-     string tipoCombustible;
27-
28- public:
29-     Automovil(string m, int a, string tc) : Vehiculo(m, a), tipoCombustible(tc) {
30-         cout << "Automóvil con combustible " << tipoCombustible << " creado." << endl;
31-     }
32-
33-     ~Automovil() {
34-         cout << "Automóvil destruido: " << marca << endl;
35-     }

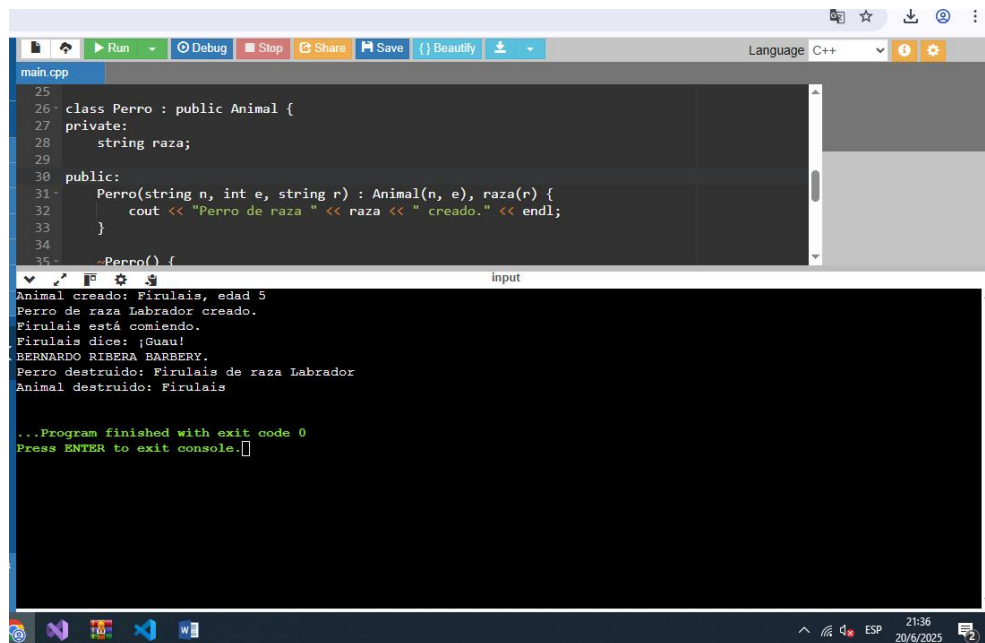
```

input

```
Vehículo creado: Toyota (2022)
Automóvil con combustible Gasolina creado.
Toyota está arrancando...
Toyota: ¡Piiii!
BERNARDO RIBERA BARBERY.
Automóvil destruido: Toyota
Vehículo destruido: Toyota

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

PERRO HERENCIA



```
main.cpp
25-
26- class Perro : public Animal {
27- private:
28-     string raza;
29-
30- public:
31-     Perro(string n, int e, string r) : Animal(n, e), raza(r) {
32-         cout << "Perro de raza " << raza << " creado." << endl;
33-     }
34-
35-     ~Perro() {

```

input

```
Animal creado: Firulaís, edad 5
Perro de raza Labrador creado.
Firulaís está comiendo.
Firulaís dice: ¡Guaú!
BERNARDO RIBERA BARBERY.
Perro destruido: Firulaís de raza Labrador
Animal destruido: Firulaís

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

