

UNIVERSIDAD PRIVADA DOMINGO SAVIO



“ACTIVIDAD”

ESTUDIANTES:

LUIS FERNANDO FELIPE SALVATIERRA MANCA

DOCENTE: REQUENA LLORENTTY JIMMY NATANIEL

MATERIA: PROGRAMACION III

CARRERA: INGENIERIA EN SISTEMAS

DEPURACION



The image shows a screenshot of a C++ IDE with a dark theme. The top toolbar includes buttons for 'Correr' (Run), 'Depurar' (Debug), 'Detener' (Stop), 'Compartir' (Share), 'Ahorrar' (Save), and 'Embellecer' (Format). The language is set to 'C++'. The code in the editor is for a program that dynamically allocates memory for an array of doubles, fills it with values, prints them, and then deallocates the memory. Comments in Spanish provide instructions and warnings. The output window at the bottom shows the program's execution, displaying the array contents and the memory deallocation message. A 'cerrar anuncio [x]' button is visible in the top right corner of the editor area.

```
principal.cpp
31 si ( p_arreglo_doubles != nullptr ) {
32     para ( int i = 0 ; i < tamaño_arreglo ; ++ i ) {
33         p_arreglo_doubles [ i ] = i * 1.5 ; // Asigna valores al arreglo
34     }
35
36     std :: cout << " Arreglo dinámico creado y llenado: " << std :: endl ;
37     para ( int i = 0 ; i < tamaño_arreglo ; ++ i ) {
38         std :: cout << " p_arreglo_doubles[ " << i << " ] = " << p_arreglo_doubles [ i ] << "
16         << " en dir: " << ( p_arreglo_doubles + i ) << std :: endl ;
17     }
18
19     eliminar [ ] p_arreglo_doubles ; //¡IMPORTANTE! Usar eliminar[] para arreglos
20     p_arreglo_doubles = nullptr ; // Buena practica
21     std :: cout << " Memoria del arreglo dinamico liberada. " << std :: endl ;
22     std :: cout << " fernando salvatierra " << std :: endl ;
23 } demás {
24     std :: cout << " ERROR: No se pudo asignar memoria para p_arreglo_doubles. " << std ::
25 }
26
27 // Intentar usar un puntero nulo (solo para demostrar, usualmente causa error o comportamiento
28 // si ( p_entero == nullptr ) {
29 // std::cout << "\np_entero es ahora nullptr." << std::endl;
30 // // *p_entero = 789; // ¡Esto causaría un error de segmentación! (Descomentar con precaución
31 // }
```

aporte

```
p_arreglo_doubles[0] = 0 en dir: 0x5fcda0c0c6e0
p_arreglo_doubles[1] = 1.5 en dir: 0x5fcda0c0c6e8
p_arreglo_doubles[2] = 3 en dir: 0x5fcda0c0c6f0
p_arreglo_doubles[3] = 4.5 en dir: 0x5fcda0c0c6f8
p_arreglo_doubles[4] = 6 en dir: 0x5fcda0c0c700
Memoria del arreglo dinamico liberada.
fernando salvatierra

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

NORBLRES MULTIPLES

```
principal.cpp
15
30 // Área de un cuadrado
11 double calcularArea ( int lado ) {
50     std :: cout << " Calculando área de CUADRADO... " << std :: endl ;
12     devolver lado * lado ;
1 }
4
58 int principal ( ) {
5     double areaCirc = calcularArea ( 5.0 ) ;           // Llama a la función de círculo
59     double areaRect = calcularArea ( 4.0 , 6.0 ) ;      // Llama a la función de rectángulo
13     double areaCuad = calcularArea ( 7 ) ;             // Llama a la función de cuadrado
2
55     std :: cout << " Área del círculo: " << areaCirc << std :: endl ;
14     std :: cout << " Área del rectángulo: " << areaRect << std :: endl ;
56     std :: cout << " Área del cuadrado: " << areaCuad << std :: endl ;
31     std :: cout << " fernando salvatierra " << std :: endl ;
57     devuelve 0 ;
3 }
```

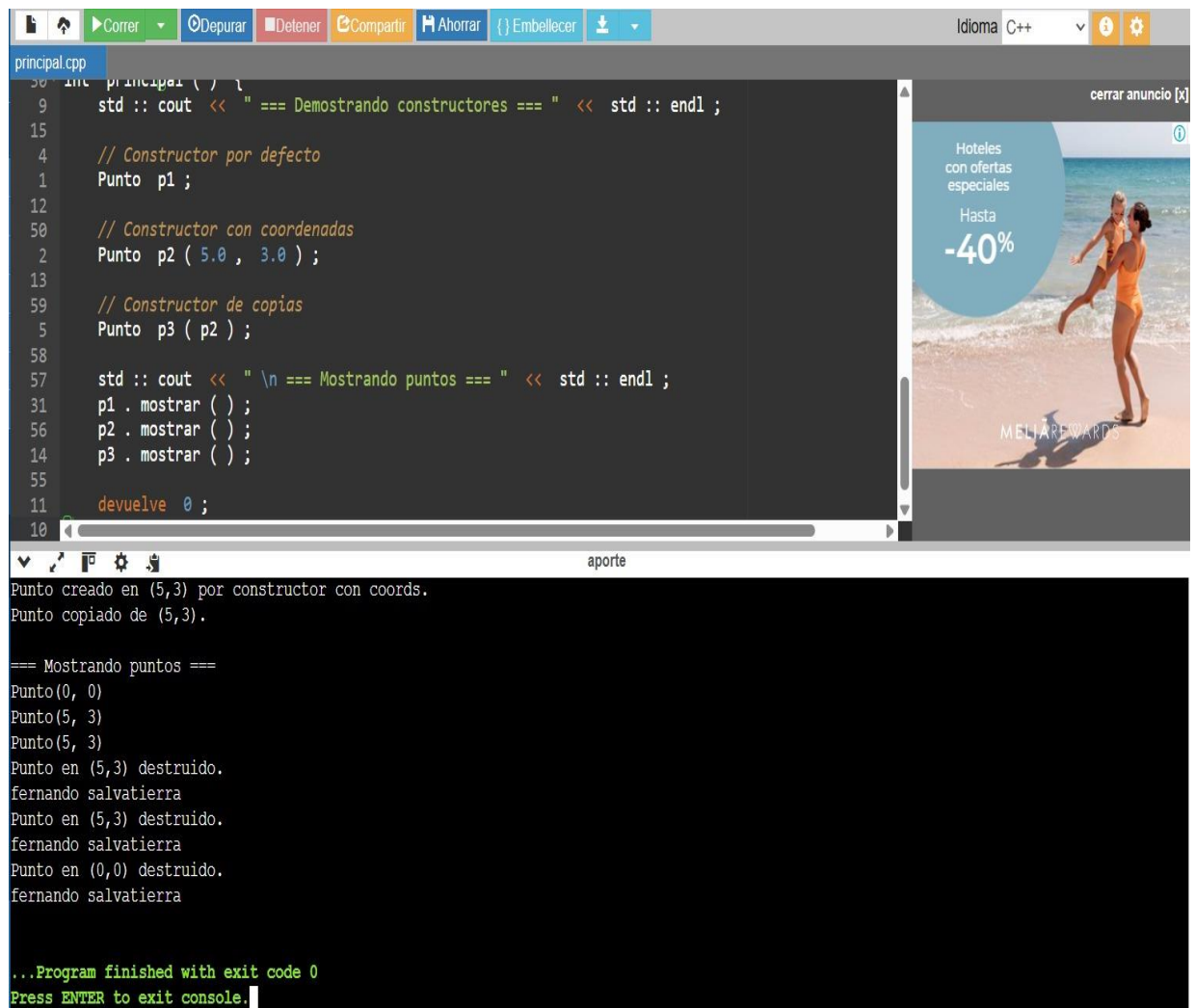
aporte

```
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
fernando salvatierra

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



MOSTRAR MULTIPLES



```
principal.cpp
30 int principal ( ) {
9     std :: cout << " === Demostrando constructores === " << std :: endl ;
15
4     // Constructor por defecto
1     Punto p1 ;
12
50    // Constructor con coordenadas
2     Punto p2 ( 5.0 , 3.0 ) ;
13
59    // Constructor de copias
5     Punto p3 ( p2 ) ;
58
57    std :: cout << " \n === Mostrando puntos === " << std :: endl ;
31    p1 . mostrar ( ) ;
56    p2 . mostrar ( ) ;
14    p3 . mostrar ( ) ;
55
11    devuelve 0 ;
10
aporte
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)
Punto en (5,3) destruido.
fernando salvatierra
Punto en (5,3) destruido.
fernando salvatierra
Punto en (0,0) destruido.
fernando salvatierra

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

MOSTRAR COMPLEMENTO

```
principal.cpp
11 std::cout << "Tipo Decimal (double): " << valor << std::endl;
12 }
1
58 void mostrar ( const std::string & valor ) {
57     std::cout << "Tipo Cadena (std::string): \" \" << valor << "\" \" << std::endl;
59     std::cout << "fernando salvatierra " << std::endl;
49 }
56
30 void mostrar ( char valor ) {
2     std::cout << "Tipo Caracter (char): ' ' << valor << "' ' << std::endl;
4 }
13
55 void mostrar ( const std::vector<int> & miVector ) {
5     std::cout << "Tipo Vector de Enteros (std::vector<int>): [ ";
31     para ( tamaño_t i = 0; i < miVector.tamaño(); ++i ) {
14         std::cout << miVector[i] << ( i == miVector.size() - 1 ? " " : ", " );
10     }
3     std::cout << "]" << std::endl;
9 }
```

cerrar anuncio [x]

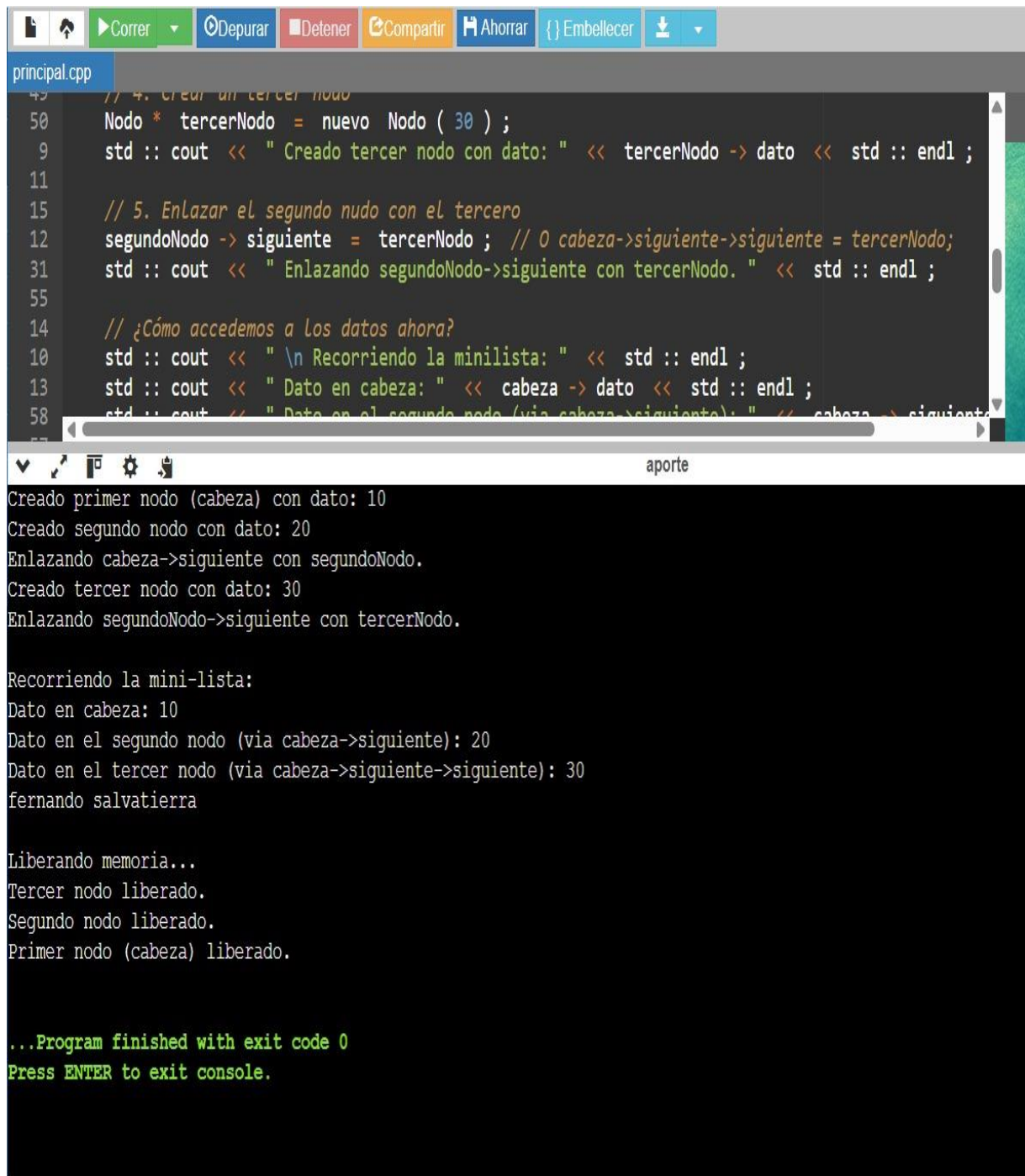
VUELVE LA
WONDER
WEEK
MELIAREWARDS

aporte

```
--- Demostracion de 'mostrar' sobrecargado ---
Tipo Entero (int): 100
Tipo Decimal (double): 3.14159
Tipo Cadena (std::string): "Hola desde Programacion III!"
fernando salvatierra
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."
fernando salvatierra

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


MINI CADENA



The image shows a C++ IDE with a file named `principal.cpp`. The code implements a linked list with three nodes. The first node is created with data 10, the second with 20, and the third with 30. The nodes are linked sequentially. The program then traverses the list, printing the data of each node. Finally, it liberates the memory for each node. The console output shows the execution of these steps.

```
principal.cpp
49 // 4. Crear un tercer nodo
50 Nodo * tercerNodo = nuevo Nodo ( 30 );
9 std :: cout << " Creado tercer nodo con dato: " << tercerNodo -> dato << std :: endl ;
11
15 // 5. Enlazar el segundo nodo con el tercero
12 segundoNodo -> siguiente = tercerNodo ; // 0 cabeza->siguiente->siguiente = tercerNodo;
31 std :: cout << " Enlazando segundoNodo->siguiente con tercerNodo. " << std :: endl ;
55
14 // ¿Cómo accedemos a los datos ahora?
10 std :: cout << " \n Recorriendo la minilista: " << std :: endl ;
13 std :: cout << " Dato en cabeza: " << cabeza -> dato << std :: endl ;
58 std :: cout << " Dato en el segundo nodo (via cabeza->siguiente): " << cabeza -> siguiente-> dato << std :: endl ;
59
```

aporte

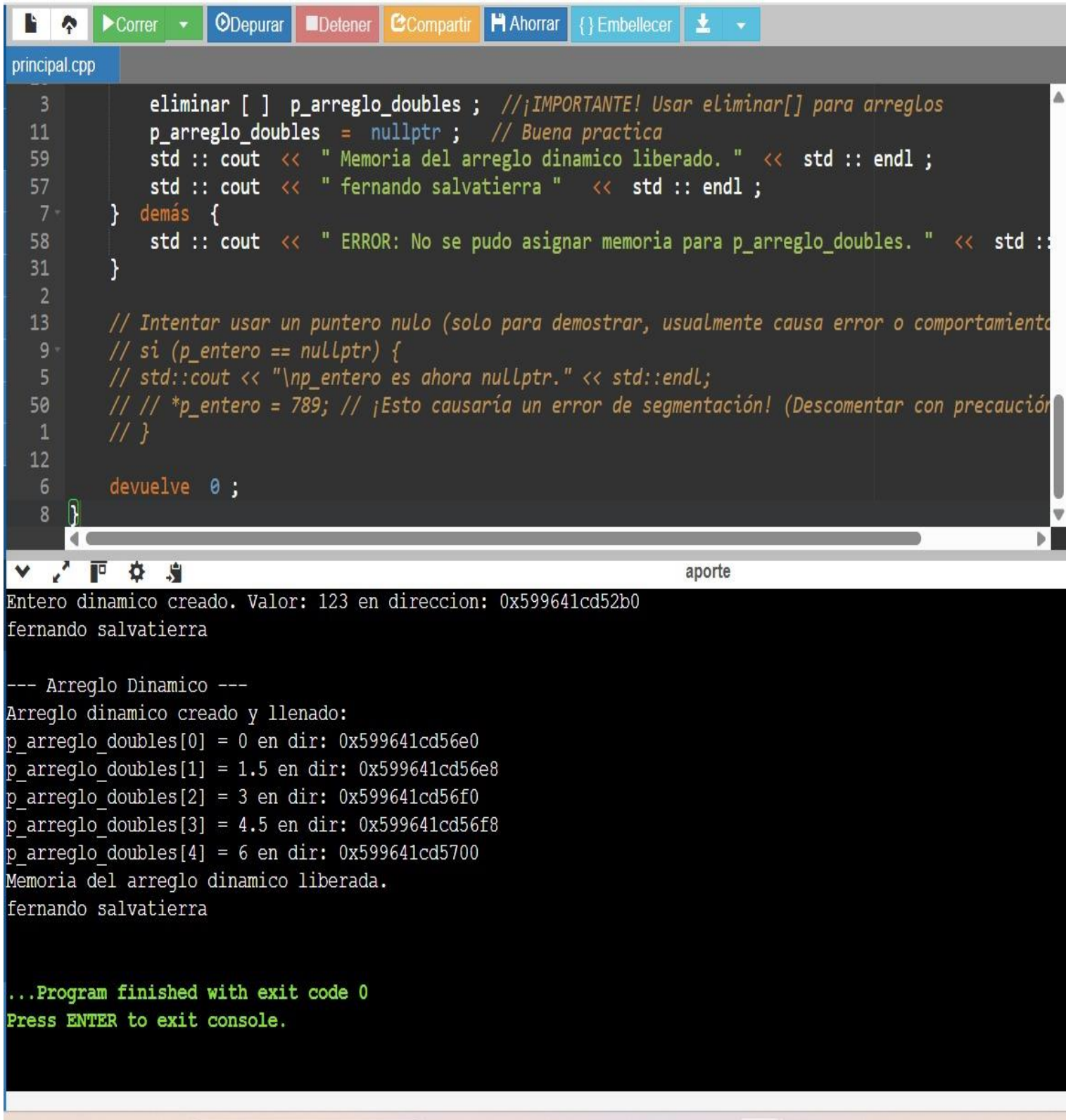
```
Creado primer nodo (cabeza) con dato: 10
Creado segundo nodo con dato: 20
Enlazando cabeza->siguiente con segundoNodo.
Creado tercer nodo con dato: 30
Enlazando segundoNodo->siguiente con tercerNodo.

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
fernando salvatierra

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

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

GESTION DINAMICA



The image shows a C++ IDE with a code editor and a console window. The code in the editor is for a program that demonstrates dynamic memory management. It includes comments in Spanish and uses `std::cout` for output. The console window shows the program's execution, including the creation and liberation of a dynamic array, and the final exit message.

```
principal.cpp
3      eliminar [ ] p_arreglo_doubles ; //¡IMPORTANTE! Usar eliminar[] para arreglos
11     p_arreglo_doubles = nullptr ; // Buena practica
59     std :: cout << " Memoria del arreglo dinamico liberado. " << std :: endl ;
57     std :: cout << " fernando salvatierra " << std :: endl ;
7     } demás {
58     std :: cout << " ERROR: No se pudo asignar memoria para p_arreglo_doubles. " << std :: endl ;
31     }
2
13     // Intentar usar un puntero nulo (solo para demostrar, usualmente causa error o comportamiento
9     // si (p_entero == nullptr) {
5     // std::cout << "\np_entero es ahora nullptr." << std::endl;
50    // // *p_entero = 789; // ¡Esto causaría un error de segmentación! (Descomentar con precaución
1     // }
12
6     devuelve 0 ;
8 }
```

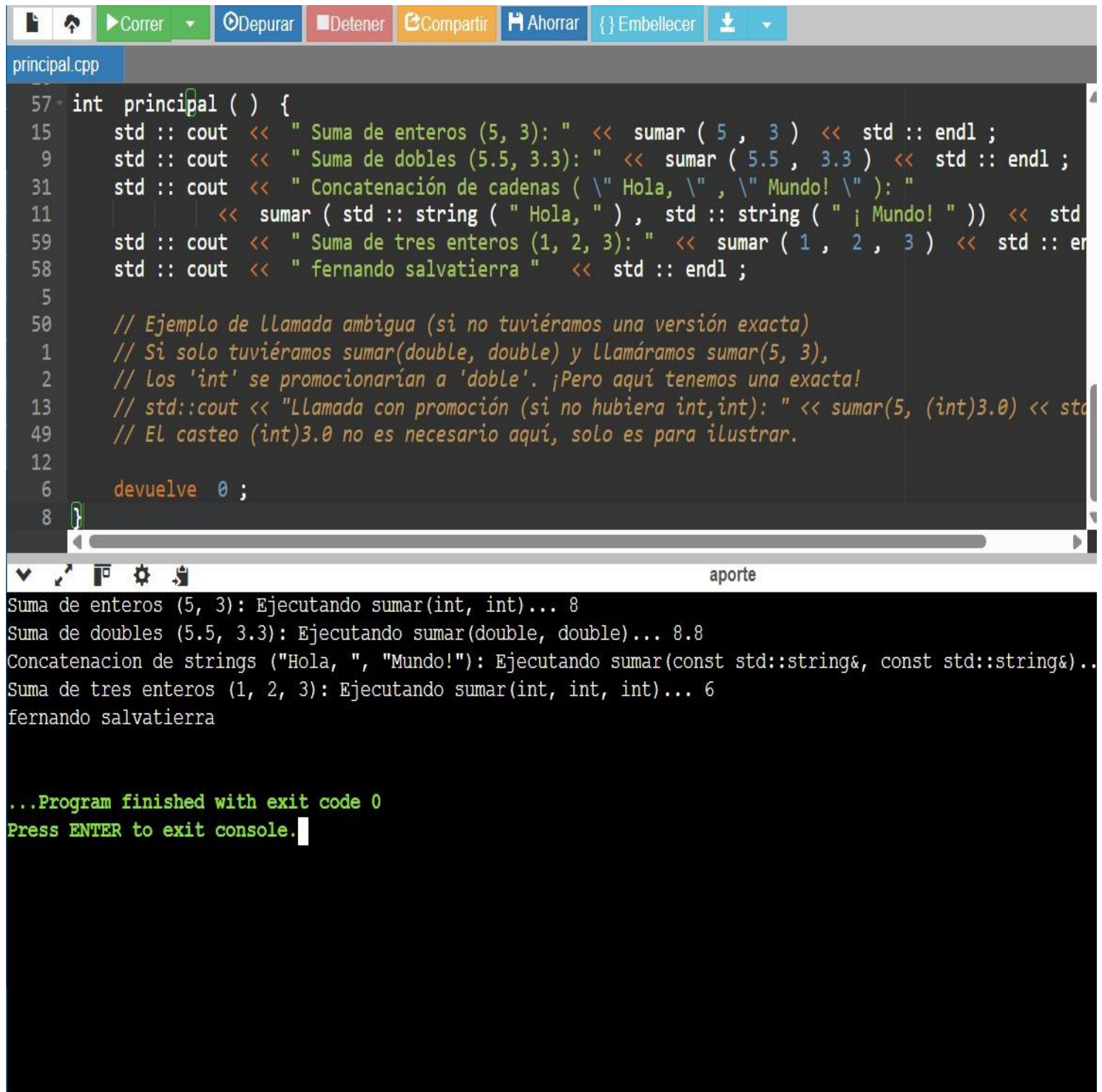
aporte

Entero dinamico creado. Valor: 123 en direccion: 0x599641cd52b0
fernando salvatierra

--- Arreglo Dinamico ---
Arreglo dinamico creado y llenado:
p_arreglo_doubles[0] = 0 en dir: 0x599641cd56e0
p_arreglo_doubles[1] = 1.5 en dir: 0x599641cd56e8
p_arreglo_doubles[2] = 3 en dir: 0x599641cd56f0
p_arreglo_doubles[3] = 4.5 en dir: 0x599641cd56f8
p_arreglo_doubles[4] = 6 en dir: 0x599641cd5700
Memoria del arreglo dinamico liberada.
fernando salvatierra

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

SOBRECARGA



The image shows a C++ IDE with a toolbar at the top containing icons for file operations, running, debugging, and other development tools. The main editor window displays a C++ file named 'principal.cpp'. The code defines a function 'sumar' with multiple overloads: for two integers, two doubles, two strings, and three integers. The 'principal' function calls these overloads and prints the results. Comments explain the concept of function overloading and how the compiler promotes integer arguments to double when only a double overload is available. The output window at the bottom shows the program's execution, displaying the results of each function call and confirming that the program finished successfully with exit code 0.

```
principal.cpp
57 int principal ( ) {
15     std :: cout << " Suma de enteros (5, 3): " << sumar ( 5 , 3 ) << std :: endl ;
9     std :: cout << " Suma de dobles (5.5, 3.3): " << sumar ( 5.5 , 3.3 ) << std :: endl ;
31     std :: cout << " Concatenación de cadenas ( \" Hola, \" , \" Mundo! \" ): "
11         << sumar ( std :: string ( " Hola, " ) , std :: string ( " ¡ Mundo! " )) << std
59     std :: cout << " Suma de tres enteros (1, 2, 3): " << sumar ( 1 , 2 , 3 ) << std :: endl ;
58     std :: cout << " fernando salvatierra " << std :: endl ;
5
50     // Ejemplo de llamada ambigua (si no tuviéramos una versión exacta)
1     // Si solo tuviéramos sumar(double, double) y llamáramos sumar(5, 3),
2     // los 'int' se promocionarían a 'doble'. ¡Pero aquí tenemos una exacta!
13     // std::cout << "Llamada con promoción (si no hubiera int,int): " << sumar(5, (int)3.0) << std
49     // El casteo (int)3.0 no es necesario aquí, solo es para ilustrar.
12
6     devuelve 0 ;
8 }
```

aporte

Suma de enteros (5, 3): Ejecutando sumar(int, int)... 8
Suma de dobles (5.5, 3.3): Ejecutando sumar(double, double)... 8.8
Concatenacion de strings ("Hola, ", "Mundo!"): Ejecutando sumar(const std::string&, const std::string&)..
Suma de tres enteros (1, 2, 3): Ejecutando sumar(int, int, int)... 6
fernando salvatierra

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

