



Taller Cpp

Estructuras

Funciones

Escuela Politécnica Nacional



02 de febrero de 2026

Ing. Yadira Franco
Rocha Asignatura:
Programación

Pinargote Javier

Contenido

Estructuras C++	4
Introducción.....	4
Desarrollo.....	4
Ejercicio 1	4
Ejercicio 2.....	5
Ejercicio 3.....	5
Ejercicio 4.....	6
Ejercicio 5.....	8
Ejercicio 6.....	9
Ejercicio 7.....	10
Ejercicio 8.....	11
Ejercicio 9.....	13
Ejercicio 10.....	13
Ejercicio 11.....	14
Ejercicio 12.....	16
Ejercicio 13.....	17
Ejercicio 14.....	19
Ejercicio 15.....	20
Ejercicio 16.....	21
Ejercicio 17.....	22
Ejercicio 18.....	24
Ejercicio 19.....	25
Ejercicio 20.....	26
Ejercicio 21.....	27
Ejercicio 22.....	28
Ejercicio 23.....	29

Ejercicio 24.....	30
Ejercicio 25.....	31
Ejercicio 26.....	32
Ejercicio 27.....	35
Ejercicio 28.....	36
Ejercicio 29.....	37
Ejercicio 30.....	38
Ejercicio 31.....	39
EJERCICIO 32: struct + vector	40
EJERCICIO 33: struct + array	41
EJERCICIO 34: struct + vector + archivo.....	42
.....	43
EJERCICIO 35: struct + vector + archivo + cálculos.....	43
EJERCICIO 36: struct + archivo + búsqueda y cálculos.....	44
EJERCICIO 37: funciones + paso por referencia + vector.....	47
EJERCICIO 38: funciones + paso por referencia + array	48
EJERCICIO 39: funciones + paso por valor + vector	49
EJERCICIO 40: funciones + paso por valor + array	50
EJERCICIO 41: EJERCICIO INTEGRADOR	51
Conclusiones.....	54
Dificultades encontradas.....	54
Repositorio GitHub.....	54

Estructuras C++

Introducción

El presente informe corresponde al Taller de C++ orientado al refuerzo del uso de estructuras condicionales, ciclos, arreglos, vectores, matrices, estructuras, funciones y manejo de archivos, con el objetivo de fortalecer la lógica de programación y el desarrollo de soluciones eficientes.

Desarrollo

Se desarrollaron 41 ejercicios en lenguaje C++, organizados en archivos independientes, abordando estructuras if, ciclos for, arreglos, vectores, matrices, menús, estructuras, funciones y archivos. Cada programa fue probado para garantizar su correcta compilación y ejecución.

Ejercicio 1

Elabore un programa en C++ que solicite un número entero e indique si el número pertenece al rango de 10 a 50, mostrando un mensaje cuando el valor sea válido y otro cuando no lo sea.

```
Ejercicio_1.cpp  X
 1  #include <iostream>
 2
 3  using namespace std;
 4
 5  int main() {
 6      int numero;
 7      cout << "Ingrese un numero entero: ";
 8      cin >> numero;
 9
10     if (numero >= 10 && numero <= 50) {
11         cout << "El numero pertenece al rango de 10 a 50." << endl;
12     } else {
13         cout << "El numero NO pertenece al rango de 10 a 50." << endl;
14     }
15
16     return 0;
17 }
```

Ilustración 1

Ilustración 2

```
El numero NO pertenece al rango de 10 a 50.
```

Ilustración 3

Ilustración 4

Ejercicio 2

Desarrolle un programa que solicite un número entero y determine si el número NO se encuentra en el rango de 10 a 50, mostrando un mensaje de advertencia cuando esté fuera del rango permitido.

```
Ejercicio_2.cpp ×
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int numero;
7     cout << "Ingrese un numero entero: ";
8     cin >> numero;
9
10    if (numero < 10 || numero > 50) {
11        cout << "ADVERTENCIA: El numero se encuentra fuera del rango permitido (10-50)." << endl;
12    } else {
13        cout << "El numero esta dentro del rango." << endl;
14    }
15
16    return 0;
17 }
```

Ilustración 7

```
C:\Users\Francisco Pinargote\ × + ▾
Ingrese un numero entero: 25
El numero esta dentro del rango.

-----
Process exited after 2.059 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 5

Ilustración 6

Ejercicio 3

Desarrolle un programa que muestre en pantalla los números del 1 al 10, pero indicando al final cuántos números fueron mostrados, utilizando únicamente la estructura for.

```
Ejercicio_3.cpp ×
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int contador = 0;
7     for (int i = 1; i <= 10; i++) {
8         cout << i << endl;
9     }
10    cout << "Se han mostrado 10 numeros." << endl;
```

Ilustración 9

Ilustración 10

```
1
2
3
4
5
6
7
8
9
10
Total de numeros mostrados: 10
-----
Process exited after 0.9841 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 11

Ilustración 12

Ejercicio 4

Realice un programa que muestre los números pares del 2 al 20 utilizando un ciclo for y muestre también la suma de dichos números.

```
Ejercicio_4.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int suma = 0;
7     for (int i = 2; i <= 20; i += 2) {
8         cout << i << endl;
9         suma += i;
10    }
11    cout << "La suma de los numeros pares es: " << suma << endl;
12    return 0;
13 }
```

Ilustración 13

Ilustración 14

```
C:\Users\Francisco Pinargote\ + ▾
2
4
6
8
10
12
14
16
18
20
La suma de los numeros pares es: 110
-----
Process exited after 1.382 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 15

Ilustración 16

Ejercicio 5

Elabore un programa que solicite un número entero positivo y muestre su tabla de multiplicar del 1 al 10, además de calcular la suma de los resultados obtenidos.

```
Ejercicio_5.cpp  X
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int numero, suma = 0;
7      cout << "Ingrese un numero entero positivo: ";
8      cin >> numero;
9
10     if (numero > 0) {
11         for (int i = 1; i <= 10; i++) {
12             int resultado = numero * i;
13             cout << numero << " x " << i << " = " << resultado << endl;
14             suma += resultado;
15         }
16         cout << "La suma de los resultados es: " << suma << endl;
17     } else {
18         cout << "El numero ingresado no es positivo." << endl;
19     }
20     return 0;
21 }
```

Ilustración 17

Ilustración 18

```
Ingrese un numero entero positivo: 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
La suma de los resultados es: 110
-----
Process exited after 3.708 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 19

Ilustración 20

Ejercicio 6

Desarrolle un programa que muestre los números del 1 al 20 e indique para cada número si es par o impar, utilizando estructuras for e if.

```
Ejercicio_6.cpp  X
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 20; i++) {
7         if (i % 2 == 0) {
8             cout << i << " es par" << endl;
9         } else {
10            cout << i << " es impar" << endl;
11        }
12    }
13    return 0;
```

Ilustración 21

Ilustración 22

```
C:\Users\Francisco Pinargote\ + ▾
1 es impar
2 es par
3 es impar
4 es par
5 es impar
6 es par
7 es impar
8 es par
9 es impar
10 es par
11 es impar
12 es par
13 es impar
14 es par
15 es impar
16 es par
17 es impar
18 es par
19 es impar
20 es par
-----
Process exited after 0.6914 seconds with return value 0
Presione una tecla para continuar . . .
```

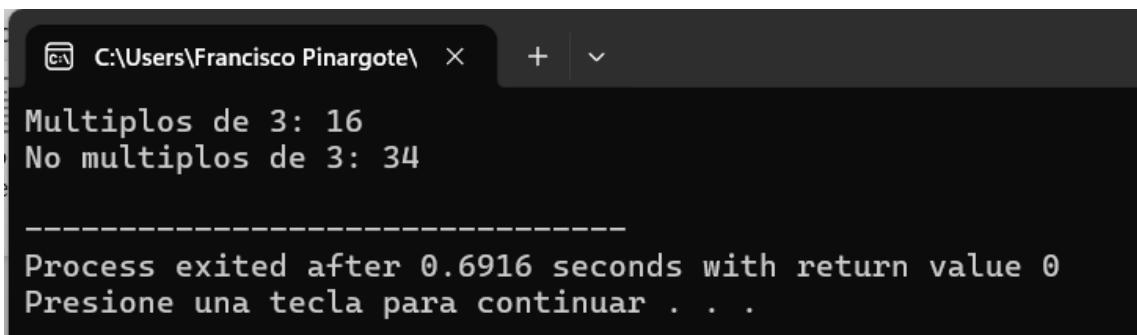
Ilustración 23

Ilustración 24

Ejercicio 7

Elabore un programa que recorra los números del 1 al 50 y cuente cuántos son múltiplos de 3 y cuántos no lo son, mostrando ambos resultados.

```
Ejercicio_7.cpp ×
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int mult3 = 0, noMult3 = 0;
7     for (int i = 1; i <= 50; i++) {
8         if (i % 3 == 0) {
9             mult3++;
10        } else {
11            noMult3++;
12        }
13    }
14    cout << "Multiplos de 3: " << mult3 << endl;
15    cout << "No Multiplos de 3: " << noMult3 << endl;
16}
```



```
C:\Users\Francisco Pinargote\ Multiplos de 3: 16
No multiplos de 3: 34

-----
Process exited after 0.6916 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 27

Ilustración 28

Ejercicio 8

Realice un programa que permita ingresar 10 números y determine cuántos son positivos, cuántos son negativos y cuántos son iguales a cero.



```
Ejercicio_8.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int num, pos = 0, neg = 0, cero = 0;
7     for (int i = 1; i <= 10; i++) {
8         cout << "Ingrese numero " << i << ": ";
9         cin >> num;
10        if (num > 0) pos++;
11        else if (num < 0) neg++;
12        else cero++;
13    }
14    cout << "Positivos: " << pos << endl;
15    cout << "Negativos: " << neg << endl;
16    cout << "Ceros: " << cero << endl;
```

Ilustración 29

Ilustración 30

The screenshot shows a terminal window with the following text output:

```
C:\Users\Francisco Pinargote\ + ▾
Ingresar numero 1: 5
Ingresar numero 2: 9
Ingresar numero 3: 7
Ingresar numero 4: 1
Ingresar numero 5: 6
Ingresar numero 6: 2
Ingresar numero 7: 7
Ingresar numero 8: 6
Ingresar numero 9: 3
Ingresar numero 10: 2
Positivos: 10
Negativos: 0
Ceros: 0
-----
Process exited after 21.07 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 31

Ilustración 32

Ejercicio 9

Desarrolle un programa que permita ingresar 15 números y determine cuántos valores se encuentran dentro del rango de 20 a 80 y cuántos están fuera de ese rango.

```
Ingrese numero 1: 58
Ingrese numero 2: 41
Ingrese numero 3: 36
Ingrese numero 4: 78
Ingrese numero 5: 45
Ingrese numero 6: 12
Ingrese numero 7: 56
Ingrese numero 8: 23
Ingrese numero 9: 89
Ingrese numero 10: 456
Ingrese numero 11: 789
Ingrese numero 12: 24
Ingrese numero 13: 25
Ingrese numero 14: 1
Ingrese numero 15: 2
Dentro del rango (20-80): 9
Fuera del rango: 6

-----
Process exited after 11.5 seconds with return value 0
Presione una tecla para continuar . . .
```

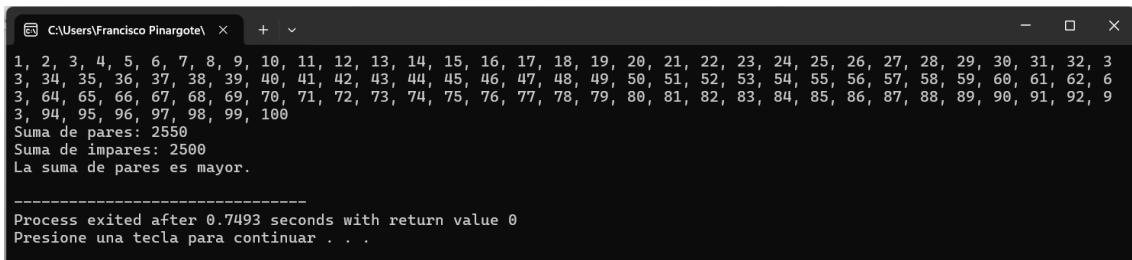
Ilustración 35

Ilustración 36

Ejercicio 10

Elabore un programa que muestre los números del 1 al 100 y calcule la suma de los números pares y la suma de los números impares, indicando al final cuál suma es mayor.

```
Ejercicio_10.cpp  X
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int sumaPares = 0, sumaImpares = 0;
7     for (int i = 1; i <= 100; i++) {
8         cout << i << (i == 100 ? "" : ", ");
9         if (i % 2 == 0) sumaPares += i;
10        else sumaImpares += i;
11    }
12    cout << "\nSuma de pares: " << sumaPares << endl;
13    cout << "Suma de impares: " << sumaImpares << endl;
14    if (sumaPares > sumaImpares) {
15        cout << "La suma de pares es mayor." << endl;
16    } else if (sumaImpares > sumaPares) {
17        cout << "La suma de impares es mayor." << endl;
18    } else {
```



```
C:\Users\Francisco Pinargote\ + 
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 3
3, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 6
3, 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, 9
3, 94, 95, 96, 97, 98, 99, 100
Suma de pares: 2550
Suma de impares: 2500
La suma de pares es mayor.

-----
Process exited after 0.7493 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 39

Ilustración 40

ARREGLOS (ARRAYS) – CON CÁLCULOS

Ejercicio 11

Desarrolle un programa que permita ingresar 10 números en un arreglo y calcule:

- La suma total de los valores
- El promedio
- El porcentaje de números positivos
- El porcentaje de números negativos

```
Ejercicio_11.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     float numeros[10], suma = 0, pos = 0, neg = 0;
7
8     for (int i = 0; i < 10; i++) {
9         cout << "Ingrese numero " << i + 1 << ": ";
10        cin >> numeros[i];
11        suma += numeros[i];
12        if (numeros[i] > 0) pos++;
13        else if (numeros[i] < 0) neg++;
14    }
15
16    cout << "Suma total: " << suma << endl;
17    cout << "Promedio: " << suma / 10 << endl;
18    cout << "Porcentaje positivos: " << (pos / 10) * 100 << "%" << endl;
19    cout << "Porcentaje negativos: " << (neg / 10) * 100 << "%" << endl;
20
21    return 0;
22 }
```

Ilustración 41

Ilustración 42

```
C:\Users\Francisco Pinargote\ + ▾  
Ingrese numero 1: 14  
Ingrese numero 2: 523  
Ingrese numero 3: 7  
Ingrese numero 4: 28  
Ingrese numero 5: 12  
Ingrese numero 6: 0  
Ingrese numero 7: 7856  
Ingrese numero 8: 87  
Ingrese numero 9: 57  
Ingrese numero 10: 57  
Suma total: 8641  
Promedio: 864.1  
Porcentaje positivos: 90%  
Porcentaje negativos: 0%  
  
-----  
Process exited after 14.27 seconds with return value 0  
Presione una tecla para continuar . . .
```

Ilustración 43

Ilustración 44

Ejercicio 12

Elabore un programa que permita ingresar 8 números en un arreglo y determine:

- El número mayor
- El número menor
- La diferencia entre el mayor y el menor

```
Ejercicio_12.cpp X  
1 #include <iostream>  
2  
3 using namespace std;  
4  
5 int main() {  
6     int numeros[8];  
7     for (int i = 0; i < 8; i++) {  
8         cout << "Ingrese numero " << i + 1 << ": ";  
9         cin >> numeros[i];  
10    }  
11  
12    int mayor = numeros[0];  
13    int menor = numeros[0];  
14  
15    for (int i = 1; i < 8; i++) {  
16        if (numeros[i] > mayor) mayor = numeros[i];  
17        if (numeros[i] < menor) menor = numeros[i];  
18    }  
19  
20    cout << "Numero mayor: " << mayor << endl;  
21    cout << "Numero menor: " << menor << endl;  
22    cout << "Diferencia: " << mayor - menor << endl;  
23  
24    return 0;  
25 }
```

```
C:\Users\Francisco Pinargote\ + | v
e
o
Ta
Ingresar numero 1: 72
Ingresar numero 2: 27
Ingresar numero 3: 2567
Ingresar numero 4: 65457
Ingresar numero 5: 17
Ingresar numero 6: 28
Ingresar numero 7: 25
Ingresar numero 8: 4
Número mayor: 65457
Número menor: 4
Diferencia: 65453
```

Ilustración 47

Ilustración 48

Ejercicio 13

Desarrolle un programa que permita ingresar 12 números en un arreglo y calcule:

- La suma de los números pares
- La suma de los números impares

- Cuál suma es mayor (pares o impares)

```
Ejercicio_13.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int numeros[12], sumaPares = 0, sumaImpares = 0;
7
8     for (int i = 0; i < 12; i++) {
9         cout << "Ingrese numero " << i + 1 << ": ";
10        cin >> numeros[i];
11        if (numeros[i] % 2 == 0) sumaPares += numeros[i];
12        else sumaImpares += numeros[i];
13    }
14
15    cout << "Suma pares: " << sumaPares << endl;
16    cout << "Suma impares: " << sumaImpares << endl;
17
18    if (sumaPares > sumaImpares) cout << "La suma de pares es mayor." << endl;
19    else if (sumaImpares > sumaPares) cout << "La suma de impares es mayor." << endl;
20    else cout << "Ambas sumas son iguales." << endl;
21
22    return 0;
23 }
```

Ilustración 49

Ilustración 50

```
Ingrese numero 1: 865
Ingrese numero 2: 74
Ingrese numero 3: 287
Ingrese numero 4: 9
Ingrese numero 5: 27
Ingrese numero 6: 742
Ingrese numero 7: 27
Ingrese numero 8:

7
Ingrese numero 9: 28
Ingrese numero 10: 8
Ingrese numero 11: 58
Ingrese numero 12: 28
Suma pares: 938
Suma impares: 1222
La suma de impares es mayor.
```

Ilustración 51

Ilustración 52

Ejercicio 14

Realice un programa que permita ingresar 10 calificaciones (0–10) en un arreglo y determine:

- El promedio general
- Cuántos estudiantes aprobaron (≥ 7)
- Cuántos reprobaron (< 7)

```
Ejercicio_14.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     float notas[10], suma = 0;
7     int aprobados = 0, reprobados = 0;
8
9     for (int i = 0; i < 10; i++) {
10         do {
11             cout << "Ingrese calificacion " << i + 1 << " (0-10): ";
12             cin >> notas[i];
13         } while (notas[i] < 0 || notas[i] > 10);
14
15         suma += notas[i];
16         if (notas[i] >= 7) aprobados++;
17         else reprobados++;
18     }
19
20     cout << "Promedio general: " << suma / 10 << endl;
21     cout << "Aprobados: " << aprobados << endl;
22     cout << "Reprobados: " << reprobados << endl;
23
24     return 0;
25 }
```

```
Ingrese calificacion 1 (0-10): 10
Ingrese calificacion 2 (0-10): 5
Ingrese calificacion 3 (0-10): 1
Ingrese calificacion 4 (0-10): 8
Ingrese calificacion 5 (0-10): 8
Ingrese calificacion 6 (0-10): 6
Ingrese calificacion 7 (0-10): 7
Ingrese calificacion 8 (0-10): 10
Ingrese calificacion 9 (0-10): 10
Ingrese calificacion 10 (0-10): 10
Promedio general: 7.5
Aprobados: 7
Reprobados: 3
```

Ilustración 55

Ilustración 56

Ejercicio 15

Elabore un programa que permita ingresar 15 números en un arreglo y calcule:

- El promedio de los números positivos
- El promedio de los números negativos

```
Ejercicio_15.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     float numeros[15], sumapos = 0, sumaneg = 0;
7     int cantPos = 0, cantNeg = 0;
8
9     for (int i = 0; i < 15; i++) {
10         cout << "Ingrese numero " << i + 1 << ": ";
11         cin >> numeros[i];
12         if (numeros[i] > 0) {
13             sumapos += numeros[i];
14             cantPos++;
15         } else if (numeros[i] < 0) {
16             sumaneg += numeros[i];
17             cantNeg++;
18         }
19     }
20
21     if (cantPos > 0) cout << "Promedio positivos: " << sumapos / cantPos << endl;
22     else cout << "No se ingresaron numeros positivos." << endl;
23
24     if (cantNeg > 0) cout << "Promedio negativos: " << sumaneg / cantNeg << endl;
```

Ilustración 57

Ilustración 58

```
C:\Users\Francisco Pinargote\ + ▾
Ingrese numero 1: 12
Ingrese numero 2: 43
Ingrese numero 3: 124
Ingrese numero 4: 73
Ingrese numero 5: 56
Ingrese numero 6: 34
Ingrese numero 7: 436
Ingrese numero 8: 9.2
Ingrese numero 9: 235
Ingrese numero 10: 235
Ingrese numero 11: 34
Ingrese numero 12: 5
Ingrese numero 13: 456
Ingrese numero 14: 87
Ingrese numero 15: 34
Promedio positivos: 124.88
No se ingresaron numeros negativos.
```

Ilustración 59

Ilustración 60

■ VECTORES – CON CÁLCULOS

Ejercicio 16

Utilizando vectores, desarrolle un programa que permita ingresar 10 números y determine:

- La suma total
- El promedio

- Cuántos números están por encima del promedio

```
Ejercicio_16.cpp [*] ×
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     vector<double> numeros(10);
7     double suma = 0;
8     int arribaPromedio = 0;
9     for (int i = 0; i < 10; i++) {
10         cout << "Ingrese numero " << i + 1 << ": ";
11         cin >> numeros[i];
12         suma += numeros[i];
13     }
14     double promedio = suma / 10;
15     for (int i = 0; i < 10; i++) {
16         if (numeros[i] > promedio) arribaPromedio++;
17     }
18     cout << "Suma total: " << suma << endl;
19     cout << "Promedio: " << promedio << endl;
20     cout << "Numeros por encima del promedio: " << arribaPromedio << endl;
21     return 0;
22 }
```

Ilustración 61

Ilustración 62

```
C:\Users\Francisco Pinargote\ × +
Ingrese numero 1: 345
Ingrese numero 2: 3463
Ingrese numero 3: 457
Ingrese numero 4: 678
Ingrese numero 5: 346
Ingrese numero 6: 57
Ingrese numero 7: 4
Ingrese numero 8: 65
Ingrese numero 9: 47
Ingrese numero 10: 36
Suma total: 5498
Promedio: 549.8
Numeros por encima del promedio: 2
```

Ilustración 63

Ilustración 64

Ejercicio 17

Elabore un programa con vectores que permita ingresar 8 valores y calcule:

- El valor máximo
- El valor mínimo

- El rango (máximo – mínimo)

```
Ejercicio_17.cpp [*] X
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     vector<int> valores(8);
7     for (int i = 0; i < 8; i++) {
8         cout << "Ingrese valor " << i + 1 << ": ";
9         cin >> valores[i];
10    }
11    int maximo = valores[0];
12    int minimo = valores[0];
13    for (int i = 1; i < 8; i++) {
14        if (valores[i] > maximo) maximo = valores[i];
15        if (valores[i] < minimo) minimo = valores[i];
16    }
17    cout << "Valor maximo: " << maximo << endl;
18    cout << "Valor minimo: " << minimo << endl;
19    cout << "Rango: " << maximo - minimo << endl;
20    return 0;0
21 }
```

Ilustración 65

Ilustración 66

```
C:\Users\Francisco Pinargote\ X
Ingrese valor 1: 23
Ingrese valor 2: 3425
Ingrese valor 3: 12
Ingrese valor 4: 4
Ingrese valor 5: 564
Ingrese valor 6:
76
Ingrese valor 7: 76
Ingrese valor 8: 3
Valor maximo: 3425
Valor minimo: 3
Rango: 3422
```

Ilustración 67

Ilustración 68

Ejercicio 18

Desarrolle un programa que permita ingresar 12 valores en un vector y determine:

- La suma de los valores positivos
- La suma de los valores negativos
- El valor absoluto total acumulado

```
Sin Nombre18 [*] X
1 #include <iostream>
2 #include <vector>
3 #include <cmath>
4 using namespace std;
5
6 int main() {
7     vector<double> valores(12);
8     double sumaPos = 0, sumaNeg = 0, absTotal = 0;
9     for (int i = 0; i < 12; i++) {
10         cout << "Ingrese valor " << i + 1 << ": ";
11         cin >> valores[i];
12         if (valores[i] > 0) sumaPos += valores[i];
13         else if (valores[i] < 0) sumaNeg += valores[i];
14         absTotal += abs(valores[i]);
15     }
16     cout << "Suma positivos: " << sumaPos << endl;
17     cout << "Suma negativos: " << sumaNeg << endl;
18     cout << "Valor absoluto total: " << absTotal << endl;
19 }
20 }
```

Ilustración 69

Ilustración 70

```

C:\Users\Francisco Pinargote\ × + ▾
Ingrese valor 1: 235
Ingrese valor 2: 342652
Ingrese valor 3: 5325
Ingrese valor 4: 235
Ingrese valor 5: 53634
Ingrese valor 6: 6235
Ingrese valor 7: 325
Ingrese valor 8: 326532
Ingrese valor 9: 5235
Ingrese valor 10: 56
Ingrese valor 11: 34
Ingrese valor 12: 23
Suma positivos: 740521
Suma negativos: 0
Valor absoluto total: 740521

```

Ilustración 71

Ilustración 72

Ejercicio 19

Utilizando vectores, elabore un programa que permita ingresar 10 edades y calcule:

- El promedio de edades
- Cuántas edades son mayores al promedio
- Cuántas edades son menores o iguales al promedio

```

Sin Nombre19 [*] ×
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     vector<int> edades(10);
7     double suma = 0;
8     int mayores = 0, menoresIguales = 0;
9     for (int i = 0; i < 10; i++) {
10         cout << "Ingrese edad " << i + 1 << ": ";
11         cin >> edades[i];
12         suma += edades[i];
13     }
14     double promedio = suma / 10;
15     for (int i = 0; i < 10; i++) {
16         if (edades[i] > promedio) mayores++;
17         else menoresIguales++;
18     }
19     cout << "Promedio de edades: " << promedio << endl;
20     cout << "Edades mayores al promedio: " << mayores << endl;
21     cout << "Edades menores o iguales al promedio: " << menoresIguales << endl;
22     return 0;
23 }

```

Ilustración 73

Ilustración 74

```
C:\Users\Francisco Pinargote\ + \ Ingrese edad 1: 23  
Ingrese edad 2: 46  
Ingrese edad 3: 65  
Ingrese edad 4: 34  
Ingrese edad 5: 565  
Ingrese edad 6: 34  
Ingrese edad 7: 56  
Ingrese edad 8: 2  
Ingrese edad 9: 45  
Ingrese edad 10: 76  
Promedio de edades: 94.6  
Edades mayores al promedio: 1  
Edades menores o iguales al promedio: 9
```

Ilustración 75

Ilustración 76

Ejercicio 20

Desarrolle un programa con vectores que permita ingresar 10 números y calcule:

- El cuadrado de cada número
- La suma de los cuadrados
- El promedio de los cuadrados

```
Ejercicio_20.cpp +  
1 #include <iostream>  
2 #include <vector>  
3 using namespace std;  
4  
5 int main() {  
6     vector<double> numeros(10);  
7     vector<double> cuadrados(10);  
8     double sumaCuadrados = 0;  
9     for (int i = 0; i < 10; i++) {  
10         cout << "Ingrese numero " << i + 1 << ": ";  
11         cin >> numeros[i];  
12         cuadrados[i] = numeros[i] * numeros[i];  
13         sumaCuadrados += cuadrados[i];  
14     }  
15     cout << "\nCuadrados de cada numero:" << endl;  
16     for (int i = 0; i < 10; i++) {  
17         cout << "Cuadrado de " << numeros[i] << " es: " << cuadrados[i] << endl;  
18     }  
19     cout << "\nSuma de los cuadrados: " << sumaCuadrados << endl;  
20     cout << "Promedio de los cuadrados: " << sumaCuadrados / 10 << endl;  
21     return 0;  
22 }
```

Ilustración 77

Ilustración 78

```
C:\Users\Francisco Pinargote\ + 
Ingrese numero 1: 1
Ingrese numero 2: 2
Ingrese numero 3: 3
Ingrese numero 4: 4
Ingrese numero 5: 5
Ingrese numero 6: 6
Ingrese numero 7: 7
Ingrese numero 8: 8
Ingrese numero 9: 9
Ingrese numero 10: 1

Cuadrados de cada numero:
Cuadrado de 1 es: 1
Cuadrado de 2 es: 4
Cuadrado de 3 es: 9
Cuadrado de 4 es: 16
Cuadrado de 5 es: 25
Cuadrado de 6 es: 36
Cuadrado de 7 es: 49
Cuadrado de 8 es: 64
Cuadrado de 9 es: 81
Cuadrado de 1 es: 1

Suma de los cuadrados: 286
Promedio de los cuadrados: 28.6
```

Ilustración 79

Ilustración 80

■ ARREGLOS / VECTORES + CONDICIONES

Ejercicio 21

Elabore un programa que permita ingresar 10 números en un arreglo y determine:

- Cuántos están en el rango de 20 a 80
- Cuántos están fuera de ese rango

- El porcentaje de cada grupo

```
Ejercicio_21.cpp [*] ×
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int numeros[10];
6     int dentro = 0,
7     for (int i = 0;
8         cout << "Ingresa un numero: ";
9         cin >> numeros[i];
10        if (numeros[i] >= 100) dentro++;
11        else fuera++;
12    }
13
14    cout << "Dentro del rango: " << dentro << endl;
15    cout << "Fuera del rango: " << fuera << endl;
16    return 0;
17 }
```

Ilustración 81

```
C:\Users\Francisco Pinargote\ × + ▾
Ingrese numero 1: 23
Ingrese numero 2: 56
Ingrese numero 3: 12324
Ingrese numero 4: 243
Ingrese numero 5: 1
Ingrese numero 6: 345
Ingrese numero 7: 67
Ingrese numero 8: 45
Ingrese numero 9: 45
Ingrese numero 10: 67
Dentro del rango: 6 (60%)
Fuera del rango: 4 (40%)
```

Ilustración 83

Ilustración 82

Ilustración 84

Ejercicio 22

Desarrolle un programa que permita ingresar 15 valores en un vector y calcule:

- Cuántos valores son múltiplos de 3
- Cuántos valores son múltiplos de 5
- Cuántos valores son múltiplos de ambos

```
Sin Nombre22 [*] ×
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     vector<int> valores(15);
7     int m3 = 0, m5 = 0, mAmbos = 0;
8     for (int i = 0; i < 15; i++) {
9         cout << "Ingrese valor " << i + 1 << ": ";
10        cin >> valores[i];
11        if (valores[i] % 3 == 0 && valores[i] % 5 == 0) mAmbos++;
12        if (valores[i] % 3 == 0) m3++;
13        if (valores[i] % 5 == 0) m5++;
14    }
15    cout << "Multiplos de 3: " << m3 << endl;
16    cout << "Multiplos de 5: " << m5 << endl;
17    cout << "Multiplos de ambos: " << mAmbos << endl;
18 }
19 }
```

Ilustración 85

Ilustración 86

```
Ingrese valor 1: 12
Ingrese valor 2: 35
Ingrese valor 3: 1234
Ingrese valor 4: 344
Ingrese valor 5: 43
Ingrese valor 6: 12
Ingrese valor 7: 436
Ingrese valor 8: 235
Ingrese valor 9: 457
Ingrese valor 10: 2
Ingrese valor 11: 2354
Ingrese valor 12: 56
Ingrese valor 13: 3
Ingrese valor 14: 34
Ingrese valor 15: 12
Multiplos de 3: 4
Multiplos de 5: 2
Multiplos de ambos: 0
```

Ilustración 87

Ilustración 88

Ejercicio 23

Elabore un programa que permita ingresar 10 precios y calcule:

- El subtotal
- El IVA (12 %)
- El total a pagar

```
Ejercicio_23.cpp  X
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float precios[10];
6     float subtotal = 0;
7     for (int i = 0; i < 10; i++) {
8         cout << "Precio " << i + 1 << ": ";
9         cin >> precios[i];
10        subtotal += precios[i];
11    }
12    float iva = subtotal * 0.12;
13    float total = subtotal + iva;
14    cout << "Subtotal: " << subtotal << endl;
15    cout << "IVA (12%): " << iva << endl;
```

Ilustración 91

Ilustración 92

```
Precio 1: 2.2
Precio 2: 3.5
Precio 3: 2.6
Precio 4: 3.1
Precio 5: 4.3
```

Ejercicio 24

Desarrolle un programa que permita ingresar 10 sueldos y calcule:

- El sueldo promedio
- Cuántos sueldos están por encima del promedio
- Cuántos están por debajo del promedio

The screenshot shows a code editor window titled "Ejercicio_24.cpp" with the following content:

```
1 // Ejercicio_24.cpp
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     vector<float> sueldos(10);
7     float suma = 0;
8     int encima = 0, debajo = 0;
9     for (int i = 0; i < 10; i++) {
10         cout << "Sueldo " << i + 1 << ": ";
11         cin >> sueldos[i];
12         suma += sueldos[i];
13     }
14     float promedio = suma / 10;
15
16     for (int i = 0; i < 10; i++) {
17         if (sueldos[i] > promedio) encima++;
18         else if (sueldos[i] < promedio) debajo++;
19     }
20     cout << "Sueldo promedio: " << promedio << endl;
21     cout << "Sueldos por encima del promedio: " << encima << endl;
22     cout << "Sueldos por debajo del promedio: " << debajo << endl;
23 }
24 }
```

Below the code editor is a terminal window titled "C:\Users\Francisco Pinargote\" showing the program's output:

```
Sueldo 1: 234
Sueldo 2: 456
Sueldo 3: 678.3
Sueldo 4: 345
Sueldo 5: 2134
Sueldo 6: 789.22
Sueldo 7: 4567
Sueldo 8: 789.3
Sueldo 9: 87878.3
Sueldo 10: 2654.25
```

Ilustración 93

Ilustración 94

■ RETO (INTEGRADOR)

Ejercicio 25

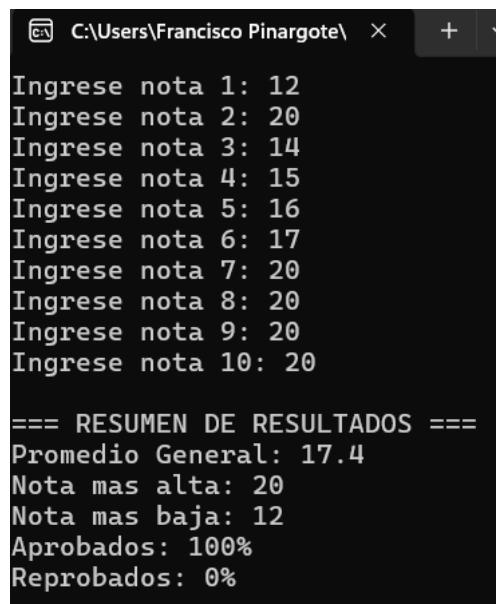
Desarrolle un programa que permita ingresar 10 notas finales y calcule:

- El promedio general
- La nota más alta y la más baja
- El porcentaje de aprobados y reprobados
- Muestre un resumen final de resultados

```
Ejercicio_25.cpp  X
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float notas[10], suma = 0, alta, baja;
6     int apr = 0, rep = 0;
7     for (int i = 0; i < 10; i++) {
8         cout << "Ingrese nota " << i + 1 << ": ";
9         cin >> notas[i];
10        suma += notas[i];
11        if (i == 0) { alta = baja = notas[i]; }
12        if (notas[i] > alta) alta = notas[i];
13        if (notas[i] < baja) baja = notas[i];
14        if (notas[i] >= 7) apr++;
15        else rep++;
16    }
17    cout << "\n==== RESUMEN DE RESULTADOS ===" << endl;
18    cout << "Promedio General: " << suma / 10 << endl;
19    cout << "Nota mas alta: " << alta << endl;
20    cout << "Nota mas baja: " << baja << endl;
21    cout << "Aprobados: " << (apr * 100.0) / 10 << "%" << endl;
22    cout << "Reprobados: " << (rep * 100.0) / 10 << "%" << endl;
23
24 }
```

Ilustración 97

Ilustración 98



```
C:\Users\Francisco Pinargote\ + \nIngrese nota 1: 12\nIngrese nota 2: 20\nIngrese nota 3: 14\nIngrese nota 4: 15\nIngrese nota 5: 16\nIngrese nota 6: 17\nIngrese nota 7: 20\nIngrese nota 8: 20\nIngrese nota 9: 20\nIngrese nota 10: 20\n\n== RESUMEN DE RESULTADOS ==\nPromedio General: 17.4\nNota mas alta: 20\nNota mas baja: 12\nAprobados: 100%\nReprobados: 0%
```

Ilustración 99

Ilustración 100

MENÚ

Ejercicio 26

Desarrolle un programa que muestre un menú con las siguientes opciones:

1. Ingresar dos números
2. Mostrar la suma y la resta
3. Mostrar el mayor de los dos números

4. Salir

```
Ejercicio_26.cpp X
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int opc;
6     float n1, n2;
7     do {
8         cout << "\n1) Ingresar numeros\n2) Suma y Resta\n3) Mostrar mayor\n4) Salir\nOpcion: ";
9         cin >> opc;
10    switch (opc) {
11        case 1: cout << "N1: "; cin >> n1; cout << "N2: "; cin >> n2; break;
12        case 2: cout << "Suma: " << n1 + n2 << "\nResta: " << n1 - n2 << endl; break;
13        case 3: cout << "Mayor: " << (n1 > n2 ? n1 : n2) << endl; break;
14    }
15    } while (opc != 4);
16    return 0;
17 }
```

Ilustración 101

Ilustración 102

```
C:\Users\Francisco Pinargote\ X

1) Ingresar numeros
2) Suma y Resta
3) Mostrar mayor
4) Salir
Opcion: 1
N1: 1
N2: 2

1) Ingresar numeros
2) Suma y Resta
3) Mostrar mayor
4) Salir
Opcion: 2
Suma: 3
Resta: -1

1) Ingresar numeros
2) Suma y Resta
3) Mostrar mayor
4) Salir
Opcion: 3
Mayor: 2

1) Ingresar numeros
2) Suma y Resta
3) Mostrar mayor
4) Salir
Opcion: |
```

Ilustración 103

Ilustración 104

Ejercicio 27

Elabore un programa que muestre un menú con las siguientes opciones:

1. Ingresar un número
2. Verificar si el número está en el rango de 1 a 100
3. Verificar si el número es par o impar
4. Salir

```
Ejercicio_27.cpp ×
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int opc, num;
6     do {
7         cout << "\n1) Ingresar numero\n2) Rango (1-100)\n3) Par o Impar\n4) Salir\nOpcion: ";
8         cin >> opc;
9         switch (opc) {
10             case 1: cout << "Numero: "; cin >> num; break;
11             case 2: cout << (num >= 1 && num <= 100 ? "En rango" : "Fuera de rango") << endl; break;
12             case 3: cout << (num % 2 == 0 ? "Par" : "Impar") << endl; break;
13         }
14     } while (opc != 4);
15     return 0;
16 }
```

Ilustración 107

```
C:\Users\Francisco Pinargote\ ×
1) Ingresar numero
2) Rango (1-100)
3) Par o Impar
4) Salir
Opcion: 1
Numero: 34

1) Ingresar numero
2) Rango (1-100)
3) Par o Impar
4) Salir
Opcion: 2
En rango

1) Ingresar numero
2) Rango (1-100)
3) Par o Impar
4) Salir
Opcion: 3
Par

1) Ingresar numero
2) Rango (1-100)
```

Ilustración 105

Ilustración 106

Ejercicio 28

Desarrolle un programa que muestre un menú con las siguientes opciones:

1. Ingresar 10 números en un arreglo
2. Mostrar el promedio
3. Mostrar el mayor y el menor
4. Contar números positivos y negativos
5. Salir

```
Ejercicio_29.cpp X
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int opc, num[10], pos, neg, mayor, menor;
6     float suma;
7     do {
8         cout << "\n1) Ingresar 10 numeros\n2) Promedio\n3) Mayor y Menor\n4) Positivos y Negativos\n";
9         cin >> opc;
10        if (opc == 1) {
11            suma = pos = neg = 0;
12            for (int i = 0; i < 10; i++) {
13                cout << "Num " << i + 1 << ": ";
14                cin >> num[i];
15                suma += num[i];
16                if (i == 0) mayor = menor = num[i];
17                if (num[i] > mayor) mayor = num[i];
18                if (num[i] < menor) menor = num[i];
19                if (num[i] > 0) pos++; else if (num[i] < 0) neg++;
20            }
21            } else if (opc == 2) cout << "Promedio: " << suma / 10 << endl;
22            else if (opc == 3) cout << "Mayor: " << mayor << " | Menor: " << menor << endl;
23            else if (opc == 4) cout << "Positivos: " << pos << " | Negativos: " << neg << endl;
24        } while (opc != 5);
25    return 0;
}
```

Ilustración 111

```
C:\Users\Francisco Pinargote\ X +
Num 4: 4
Num 5: 5
Num 6: 6
Num 7: 7
Num 8: 8
Num 9: 9
Num 10: 11

1) Ingresar 10 numeros
2) Promedio
3) Mayor y Menor
4) Positivos y Negativos
5) Salir
Opcion: 2
Promedio: 5.6

1) Ingresar 10 numeros
2) Promedio
3) Mayor y Menor
4) Positivos y Negativos
5) Salir
Opcion: 3
Mayor: 11 | Menor: 1
```

Ilustración 109

Ilustración 110

F. MATRICES

Ejercicio 29

Elabore un programa que permita ingresar una matriz 3x3 y determine:

- La suma de cada fila
- La suma total de la matriz

```
Ejercicio_28.cpp  X
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int mat[3][3], total = 0;
6     for (int i = 0; i < 3; i++) {
7         int filaSuma = 0;
8         for (int j = 0; j < 3; j++) {
9             cout << "Matriz[" << i << "][" << j << "]: "; cin >> mat[i][j];
10            filaSuma += mat[i][j];
11            total += mat[i][j];
12        }
13        cout << "Suma fila " << i << ": " << filaSuma << endl;
14    }
15    cout << "Suma total: " << total << endl;
16    return 0;
17 }
```

Ilustración 115

Ilustración 116

```
C:\Users\Francisco Pinargote\ >
Matriz[0][0]: 1
Matriz[0][1]: 2
Matriz[0][2]: 3
Suma fila 0: 6
Matriz[1][0]: 4
Matriz[1][1]: 5
Matriz[1][2]: 6
Suma fila 1: 15
Matriz[2][0]: 7
Matriz[2][1]: 8
Matriz[2][2]: 9
Suma fila 2: 24
Suma total: 45
```

Ilustración 113

Ilustración 114

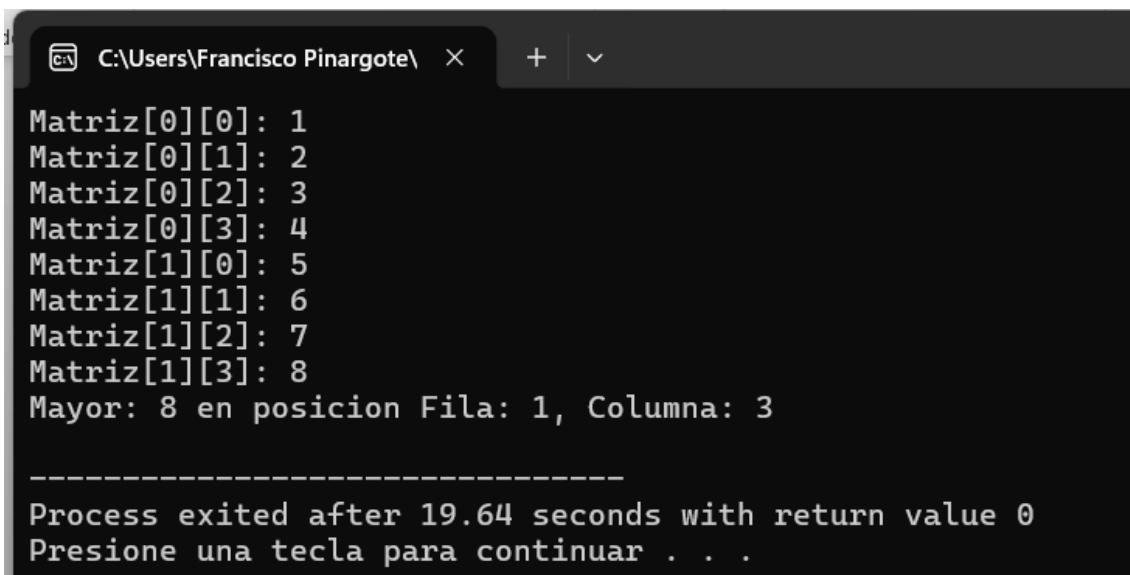
Ejercicio 30

Desarrolle un programa que permita ingresar una matriz 2x4 y determine:

- El número mayor
- La posición (fila y columna) del número mayor

```
Ejercicio_30.cpp x
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int mat[2][4], mayor, f, c;
6     for (int i = 0; i < 2; i++) {
7         for (int j = 0; j < 4; j++) {
8             cout << "Matriz[" << i << "][" << j << "]: ";
9             cin >> mat[i][j];
10            if (i == 0 && j == 0) { mayor = mat[i][j]; f = i; c = j; }
11            if (mat[i][j] > mayor) { mayor = mat[i][j]; f = i; c = j; }
12        }
13    cout << "Mayor: " << mayor << " en posicion Fila: " << f << ", Columna: " << c << endl;
14    return 0;
15 }
```

Ilustración 119



```
C:\Users\Francisco Pinargote\ x + v
Matriz[0][0]: 1
Matriz[0][1]: 2
Matriz[0][2]: 3
Matriz[0][3]: 4
Matriz[1][0]: 5
Matriz[1][1]: 6
Matriz[1][2]: 7
Matriz[1][3]: 8
Mayor: 8 en posicion Fila: 1, Columna: 3
-----
Process exited after 19.64 seconds with return value 0
Presione una tecla para continuar . . .
```

Ilustración 117

Ilustración 118

Ejercicio 31

Realice un programa que permita ingresar una matriz 4x4 y cuente cuántos valores son:

- Positivos
- Negativos
- Iguales a cero

The screenshot shows a code editor window titled "Ejercicio_31.cpp" with the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int mat[4][4], pos = 0, neg = 0, cero = 0;
6     for (int i = 0; i < 4; i++) {
7         for (int j = 0; j < 4; j++) {
8             cout << "Matriz[" << i << "][" << j << "]": "; cin >> mat[i][j];
9             if (mat[i][j] > 0) pos++;
10            else if (mat[i][j] < 0) neg++;
11            else cero++;
12        }
13    }
14    cout << "\nPositivos: " << pos << endl;
15    cout << "\nNegativos: " << neg << endl;
16    cout << "\nCeros: " << cero << endl;
17
18    return 0;
19}
```

The terminal window below shows the execution of the program. It prompts for a 4x4 matrix and then displays the count of positive, negative, and zero values.

Ilustración 12

```
C:\Users\Francisco Pinargote\
```

Matriz[0][0]: 11
Matriz[0][1]: 12
Matriz[0][2]: 13
Matriz[0][3]: 14
Matriz[1][0]: 15
Matriz[1][1]: 56
Matriz[1][2]: 67
Matriz[1][3]: 78
Matriz[2][0]: 89
Matriz[2][1]: 09
Matriz[2][2]: 34
Matriz[2][3]: 46
Matriz[3][0]: 67
Matriz[3][1]: 45
Matriz[3][2]: 567
Matriz[3][3]: 78

Ilustración 12

Ilustración 12

Ilustración 121

Ilustración 122

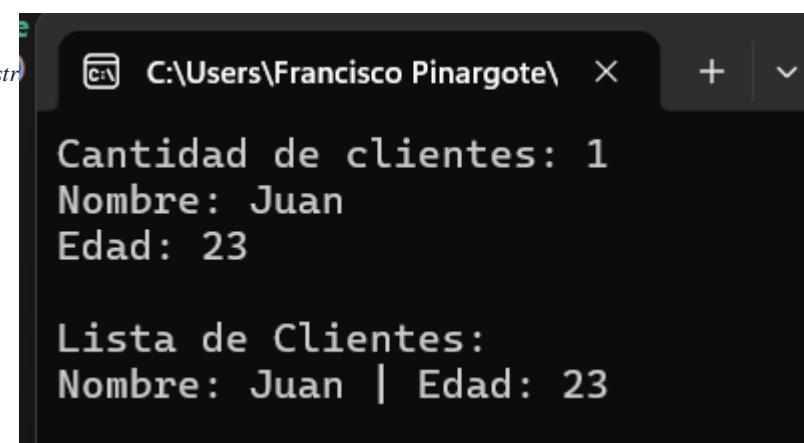
EJERCICIO 32: struct + vector

Enunciado:

Desarrolle un programa en C++ que defina una estructura Cliente con los campos nombre y edad. El programa debe permitir ingresar varios clientes usando un vector y luego mostrar en pantalla la información de todos los clientes registrados.

```
Sin Nombre32 [*] ×
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5
6  struct Cliente {
7      string nombre;
8      int edad;
9  };
10
11 int main() {
12     vector<Cliente> clientes;
13     int n;
14     cout << "Cantidad de clientes: "; cin >> n;
15     for (int i = 0; i < n; i++) {
16         Cliente c;
17         cin.ignore();
18         cout << "Nombre: "; getline(cin, c.nombre);
19         cout << "Edad: "; cin >> c.edad;
20         clientes.push_back(c);
21     }
22     cout << "\nLista de Clientes:" << endl;
23     for (const auto& c : clientes) {
24         cout << "Nombre: " << c.nombre << " | Edad: " << c.edad << endl;
25     }
26     return 0;
27 }
```

Ilustración 127



```
Cantidad de clientes: 1
Nombre: Juan
Edad: 23

Lista de Clientes:
Nombre: Juan | Edad: 23
```

Ilustración 125

Ilustración 126

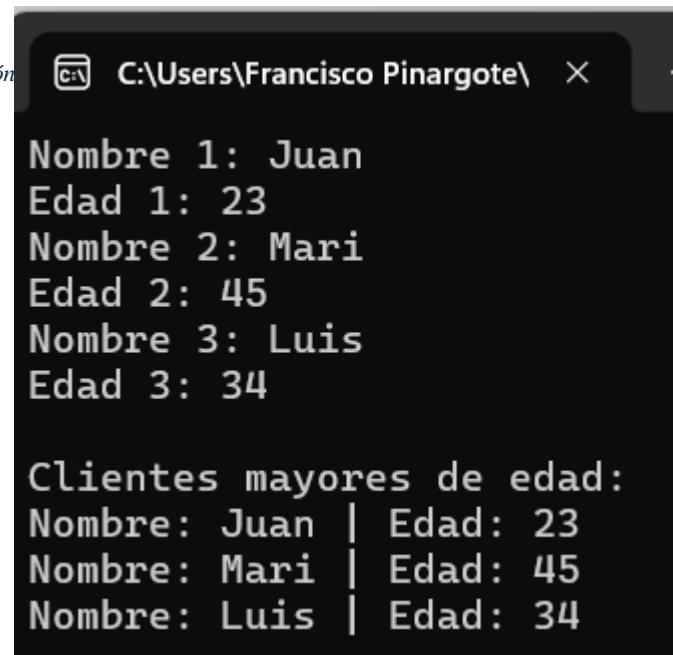
EJERCICIO 33: struct + array

Enunciado:

Cree un programa en C++ que utilice una estructura Cliente con los campos nombre y edad. Almacene los datos de tres clientes en un arreglo y muestre únicamente los clientes que sean mayores de edad (edad mayor o igual a 18).

```
Sin Nombre33 [*] ×
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Cliente {
6     string nombre;
7     int edad;
8 };
9
10 int main() {
11     Cliente clientes[3];
12     for (int i = 0; i < 3; i++) {
13         cout << "Nombre " << i + 1 << ": "; cin >> clientes[i].nombre;
14         cout << "Edad " << i + 1 << ": "; cin >> clientes[i].edad;
15     }
16     cout << "\nClientes mayores de edad:" << endl;
17     for (int i = 0; i < 3; i++) {
18         if (clientes[i].edad >= 18) {
19             cout << "Nombre: " << clientes[i].nombre << " | Edad: " << clientes[i].edad << endl;
20         }
21     }
22 }
23 }
```

Ilustración 131



```
Ilustración 129 C:\Users\Francisco Pinargote\ ×

Nombre 1: Juan
Edad 1: 23
Nombre 2: Mari
Edad 2: 45
Nombre 3: Luis
Edad 3: 34

Clientes mayores de edad:
Nombre: Juan | Edad: 23
Nombre: Mari | Edad: 45
Nombre: Luis | Edad: 34
```

Ilustración 129

Ilustración 130

EJERCICIO 34: struct + vector + archivo

Enunciado:

Realice un programa en C++ que use una estructura Cliente con los campos nombre y edad. El programa debe almacenar los datos en un vector, guardarlos en un archivo de texto y permitir buscar un cliente por su nombre, mostrando su edad si se encuentra registrado.

```
Sin Nombre34 [*] ×
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <fstream>
5
6  using namespace std;
7
8  struct Cliente {
9      string nombre;
10     int edad;
11 };
12
13 int main() {
14     vector<Cliente> clientes;
15     int n;
16     cout << "Cantidad: "; cin >> n;
17     ofstream archivo("clientes.txt");
18     for (int i = 0; i < n; i++) {
19         Cliente c;
20         cin.ignore();
21         cout << "Nombre: "; getline(cin, c.nombre);
22         cout << "Edad: "; cin >> c.edad;
23         clientes.push_back(c);
24         archivo << c.nombre << " " << c.edad << endl;
25     }
26     archivo.close();
27     string buscar;
28     cout << "\nNombre a buscar: "; cin.ignore(); getline(cin, buscar);
```

```
Ilustración 133
C:\Users\Francisco Pinargote\ × +
Ilustración 133
Cantidad: 2
Nombre: Juan
Edad: 23
Nombre: Maria
Edad: 45
```

Ilustración 133

Ilustración 134

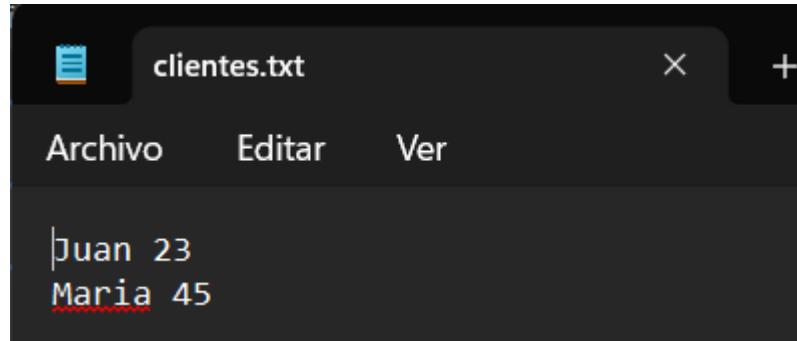


Ilustración 137

Ilustración 138

EJERCICIO 35: struct + vector + archivo + cálculos

Enunciado:

Desarrolle un programa en C++ que defina una estructura Cliente con los campos nombre, edad y consumoMensual. El programa debe almacenar los datos en un vector, guardarlos en un archivo, leerlos posteriormente y calcular el consumo total, el consumo promedio y el cliente con mayor consumo.

```
Sin Nombre35 [*] ×
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <fstream>
5
6 using namespace std;
7
8 struct Cliente {
9     string nombre;
10    int edad;
11    double consumoMensual;
12 };
13
14 int main() {
15     vector<Cliente> clientes;
16     int n;
17     cout << "Cantidad: "; cin >> n;
18
19     ofstream salida("consumos.txt");
20     for (int i = 0; i < n; i++) {
21         Cliente c;
22         cin.ignore();
23         cout << "Nombre: "; getline(cin, c.nombre);
24         cout << "Edad: "; cin >> c.edad;
25         cout << "Consumo: "; cin >> c.consumoMensual;
26         clientes.push_back(c);
27         salida << c.nombre << " " << c.edad << " " << c.consumoMensual << endl;
28     }
}
```

Ilustración 139

Ilustración 140

```
C:\Users\Francisco Pinargote\ + v
Cantidad: 2
Nombre: maria
Edad: 23
Consumo: 45.8
Nombre: juan
Edad: 34
Consumo: 100.64

Total: 146.44
Promedio: 73.22
Mayor consumo: juan (100.64)
```

Ilustración 143



Ilustración 141

Ilustración 142

EJERCICIO 36: struct + archivo + búsqueda y cálculos

Enunciado:

Cree un programa en C++ que utilice una estructura Estudiante con los campos nombre, nota1, nota2 y nota3. El programa debe guardar los datos en un archivo, leerlos, calcular el promedio de cada estudiante y permitir buscar un estudiante por nombre para mostrar sus notas, su promedio y si aprueba o reprueba (promedio mayor o igual a 7).


```
Sin Nombre36 [*] ×
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <fstream>
5
6 using namespace std;
7
8 struct Estudiante {
9     string nombre;
10    float n1, n2, n3;
11 };
12
13 int main() {
14     int n;
15     cout << "Cantidad: "; cin >> n;
16     ofstream arch("notas.txt");
17     for (int i = 0; i < n; i++) {
18         Estudiante e;
19         cout << "Nombre: "; cin >> e.nombre;
20         cout << "Notas 1, 2 y 3: "; cin >> e.n1 >> e.n2 >> e.n3;
21         arch << e.nombre << " " << e.n1 << " " << e.n2 << " "
22     }
23     arch.close();
24
25     string b;
26     cout << "Buscar nombre: "; cin >> b;
27     ifstream lectura("notas.txt");
28     string nom; float v1, v2, v3;
```

Ilustración 149

```
C:\Users\Francisco Pinargote\ × + | ▾
Cantidad: 2
Nombre: juan
Notas 1, 2 y 3: 12
23
345
Nombre: maria
Notas 1, 2 y 3: 12
34
45
Buscar nombre: maria
Notas: 12, 34, 45
Promedio: 30.3333 | Aprueba
```

Ilustración 147

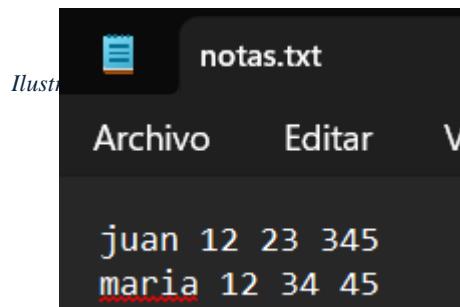


Ilustración 146

EJERCICIO 37: funciones + paso por referencia + vector

Enunciado:

Desarrolle un programa en C++ que utilice funciones con paso de parámetros por referencia y un vector de números enteros. El programa debe incluir funciones para llenar el vector y para calcular la suma, el promedio, el valor mayor y el valor menor de los elementos almacenados.

```
Sin Nombre37 [*] X
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 void llenar(vector<int>& v, int n) {
7     for (int i = 0; i < n; i++) {
8         int val; cout << "Valor " << i + 1 << ": ";
9         v.push_back(val);
10    }
11 }
12
13 void calcular(vector<int>& v, int& s, float& p, in
14     s = 0; ma = v[0]; me = v[0];
15     for (int x : v) {
16         s += x;
17         if (x > ma) ma = x;
18         if (x < me) me = x;
19     }
20     p = (float)s / v.size();
21 }
22
23 int main() {
24     vector<int> vec;
25     int n, suma, mayor, menor; float prom;
26     cout << "N: "; cin >> n;
27     llenar(vec, n);
28     calcular(vec, suma, prom, mayor, menor);
```

Ilustración 153

Ilustración 154

```
C:\Users\Francisco Pinard
N: 7
Valor 1: 12
Valor 2: 1
Valor 3: 2
Valor 4: 34
Valor 5: 45
Valor 6: 6
Valor 7: 7
Suma: 107
```

Ilustración 151

Ilustración 152

EJERCICIO 38: funciones + paso por referencia + array

Enunciado:

Cree un programa en C++ que utilice un arreglo de números reales y funciones con paso por referencia. El programa debe llenar el arreglo mediante una función y calcular el total de ventas, el promedio de ventas y cuántas ventas son mayores al promedio.

```
Sin Nombre38 [M] ×
1 #include <iostream>
2
3 using namespace std;
4
5 void llenar(float a[], int n) {
6     for (int i = 0; i < n; i++) {
7         cout << "Venta " << i + 1 << ": "; cin >> a[i];
8     }
9 }
10
11 void procesar(float a[], int n, float& t, float& p, int& m) {
12     t = 0; m = 0;
13     for (int i = 0; i < n; i++) t += a[i];
14     p = t / n;
15     for (int i = 0; i < n; i++) if (a[i] > p) m++;
16 }
17
18 int main() {
19     float v[8], total, promedio; int mayores;
20     llenar(v, 8);
21     procesar(v, 8, total, promedio, mayores);
22     cout << "Total: " << total << "\nPromedio: " << promedio << "\nMayores al prom: "
23     return 0;
24 }
```

Ilustración 157

```
Ilustración 158
C:\Users\Francisco Pinargote\ ×
Venta 1: 23
Venta 2: 435
Venta 3: 1.5
Venta 4: 55854.6
Venta 5: 26
Venta 6: 21
Venta 7: 0
Venta 8: 3.9
Total: 56365
Promedio: 7045.62
Mayores al prom: 1
```

Ilustración 155

Ilustración 156

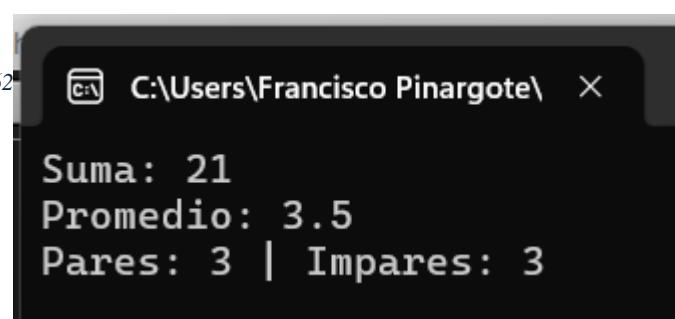
EJERCICIO 39: funciones + paso por valor + vector

Enunciado:

Desarrolle un programa en C++ que utilice funciones con paso de parámetros por valor y un vector de números enteros. El programa debe pasar el vector a las funciones para calcular la suma de los elementos, el promedio y determinar cuántos valores son pares e impares, sin modificar el contenido original del vector.

```
Sin Nombre39 [*] X
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int suma(vector<int> v) {
7     int s = 0;
8     for (int x : v) s += x;
9     return s;
10 }
11
12 void paresImpares(vector<int> v, int& p, int& i) {
13     p = 0; i = 0;
14     for (int x : v) if (x % 2 == 0) p++; else i++;
15 }
16
17 int main() {
18     vector<int> vec = {1, 2, 3, 4, 5, 6};
19     int p, i;
20     cout << "Suma: " << suma(vec) << endl;
21     cout << "Promedio: " << (float)suma(vec) / vec.size() << endl;
22     paresImpares(vec, p, i);
23     cout << "Pares: " << p << " | Impares: " << i << endl;
24     return 0;
25 }
```

Ilustración 161



```
C:\Users\Francisco Pinargote\ X
Suma: 21
Promedio: 3.5
Pares: 3 | Impares: 3
```

Ilustración 159

Ilustración 160

EJERCICIO 40: funciones + paso por valor + array

Enunciado:

Cree un programa en C++ que utilice un arreglo de números enteros y funciones con paso por valor. El programa debe enviar el arreglo a las funciones para calcular el valor mayor, el valor menor y la cantidad de números positivos, negativos y ceros, sin alterar los datos originales.

```
Sin Nombre40 [*] x
1 #include <iostream>
2
3 using namespace std;
4
5 int obtenerMayor(int arr[], int n) {
6     int mayor = arr[0];
7     for (int i = 1; i < n; i++) {
8         if (arr[i] > mayor) mayor = arr[i];
9     }
10    return mayor;
11}
12
13 int obtenerMenor(int arr[], int n) {
14     int menor = arr[0];
15     for (int i = 1; i < n; i++) {
16         if (arr[i] < menor) menor = arr[i];
17     }
18    return menor;
19}
20
21 void contarTipos(int arr[], int n, int& pos, int& neg, int& ceros) {
22     pos = 0; neg = 0; ceros = 0;
23     for (int i = 0; i < n; i++) {
24         if (arr[i] > 0) pos++;
25         else if (arr[i] < 0) neg++;
26         else ceros++;
27     }
28 }
```

Ilustración

```
Ingrese numero 1: 23
Ingrese numero 2: 43
Ingrese numero 3: 0
Ingrese numero 4: 235
Ingrese numero 5: 35
Ingrese numero 6: 56
Ingrese numero 7: 34
Ingrese numero 8: 56
Ingrese numero 9: 32
Ingrese numero 10: 45
```

Ilustración 163

Ilustración 164

EJERCICIO 41: EJERCICIO INTEGRADOR

Enunciado:

Desarrolle un programa en C++ que permita gestionar las ventas diarias de una tienda.

El programa 41.1 CALIFICACIONES DE UN ESTUDIANTE

Enunciado:

Desarrolle un programa en C++ que permita gestionar las calificaciones de un estudiante.

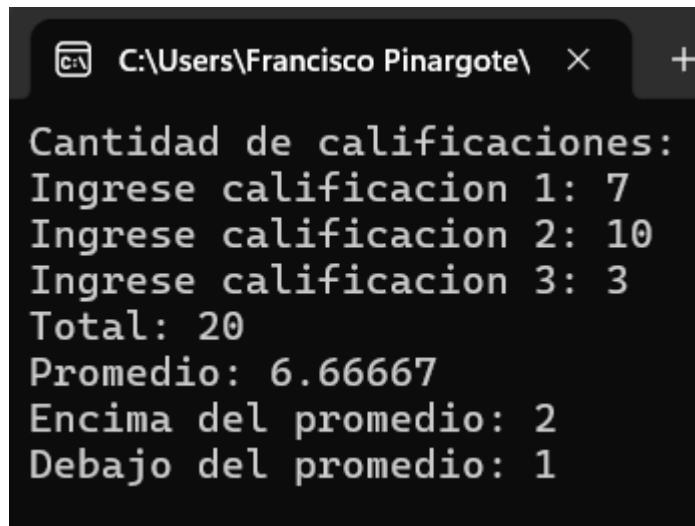
El programa debe cumplir con los siguientes requisitos:

- Usar un vector de números reales para almacenar las calificaciones.
- Utilizar funciones para ingresar las calificaciones y realizar los cálculos.
- Guardar las calificaciones en un archivo de texto.
- Leer los datos desde el archivo.
- Calcular el total de calificaciones y el promedio.
- Mostrar cuántas calificaciones están por encima del promedio y cuántas están por debajo del promedio.
- Evitar repetir cálculos innecesarios.

```
Sin Nombre41 [*] ×
 1  #include <iostream>
 2  #include <vector>
 3  #include <fstream>
 4  #include <numeric>
 5
 6  using namespace std;
 7
 8  void ingresarCalificaciones(vector<double>& notas, int n) {
 9    for (int i = 0; i < n; i++) {
10      double nota;
11      cout << "Ingrese calificación " << i + 1 << ": ";
12      cin >> nota;
13      notas.push_back(nota);
14    }
15  }
16
17  void guardarArchivo(const vector<double>& notas) {
18    ofstream archivo("calificaciones.txt");
19    for (double n : notas) {
20      archivo << n << endl;
21    }
22    archivo.close();
23  }
24
25  void leerArchivo(vector<double>& notas) {
26    notas.clear();
27    ifstream archivo("calificaciones.txt");
28    double nota;
```

Ilustración 167

Ilustración 168



```
C:\Users\Francisco Pinargote\ +  
Cantidad de calificaciones:  
Ingrese calificacion 1: 7  
Ingrese calificacion 2: 10  
Ingrese calificacion 3: 3  
Total: 20  
Promedio: 6.66667  
Encima del promedio: 2  
Deabajo del promedio: 1
```

Ilustración 169

Ilustración 170

El programa 41.2: VENTAS

El programa debe desarrollarse en C++ y compilar correctamente.

Se debe utilizar un vector<double> para ingresar y almacenar inicialmente los valores de las ventas.

El ingreso de las ventas debe realizarse mediante una función que reciba el vector por referencia.

Una vez ingresados los datos, las ventas deben guardarse en un archivo de texto llamado ventas.txt.

El programa debe leer las ventas desde el archivo y almacenar esos valores en un nuevo vector<double>.

Todos los cálculos deben realizarse únicamente con el vector cargado desde el archivo.

Se deben usar funciones para:

- Calcular el total de ventas.
- Calcular el promedio de ventas (este cálculo debe hacerse una sola vez).

Utilizando el promedio calculado, el programa debe determinar:

- Cuántas ventas están por encima del promedio.
- Cuántas ventas están por debajo del promedio.

El programa debe mostrar en pantalla:

- Total de ventas.

- Promedio de ventas.
- Cantidad de ventas mayores y menores al promedio.

No se permite repetir cálculos innecesarios ni recalcular el promedio dentro de ciclos.

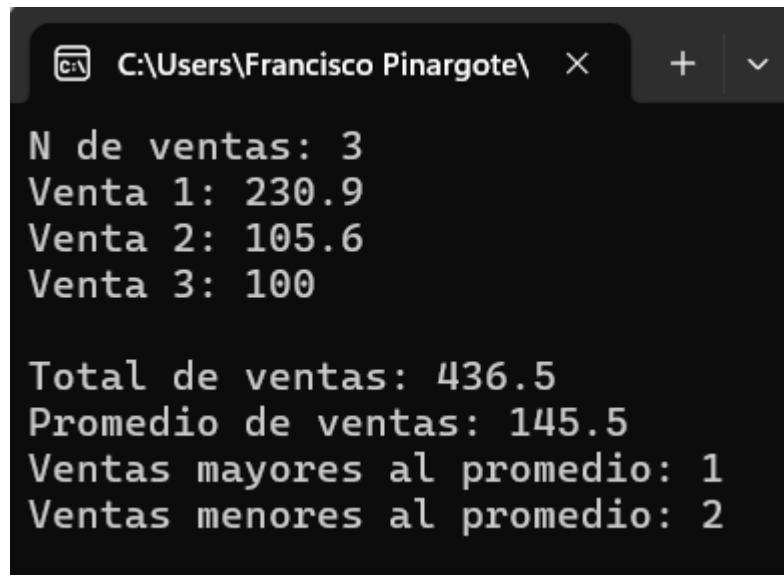
No se permite el uso de variables globales.

El código debe estar correctamente indentado y comentado.

```
Ejercicio_41.2.cpp ^
1 #include <iostream>
2 #include <vector>
3 #include <fstream>
4
5 using namespace std;
6
7 void ingresarVentas(vector<double>& v, int n) {
8     for (int i = 0; i < n; i++) {
9         double venta;
10        cout << "Venta " << i + 1 << ": ";
11        cin >> venta;
12        v.push_back(venta);
13    }
14 }
15
16 void guardarVentas(const vector<double>& v) {
17     ofstream arch("ventas.txt");
18     for (double x : v) arch << x << endl;
19     arch.close();
20 }
21
22 void cargarVentas(vector<double>& v) {
23     v.clear();
24     ifstream arch("ventas.txt");
25     double x;
26     while (arch >> x) v.push_back(x);
27     arch.close();
```

Ilustración 171

Ilustración 172



A screenshot of a terminal window titled 'C:\Users\Francisco Pinargote\'. The window displays the following text output:

```
N de ventas: 3
Venta 1: 230.9
Venta 2: 105.6
Venta 3: 100

Total de ventas: 436.5
Promedio de ventas: 145.5
Ventas mayores al promedio: 1
Ventas menores al promedio: 2
```

Ilustración 173

Conclusiones

El taller permitió reforzar conceptos fundamentales de programación estructurada, mejorar el uso correcto de operadores lógicos y relacionales, y afianzar el manejo de estructuras de datos y funciones en C++.

Dificultades encontradas

Durante el desarrollo del taller se presentaron dificultades en la correcta aplicación de operadores lógicos y en la organización del código, las cuales fueron superadas mediante práctica y revisión constante.

Repositorio GitHub

. https://github.com/Javierprgt/Taller_Cpp_Estructuras_Funciones.git