

## ARRAYS EN C/C++

---

- *Profesor: York Mansilla*
- *Lenguaje: C/C++*
- *IDE: Visual Studio Code*
- *Compilador: MSYS2*





# Proceso de instalacion Visual Studio Code

*Primero instalar el Visual Studio Code por la página oficial, luego decidí configurarlo a gusto, para finalmente instalar el compilador MSYS2. Los links que utilicé son:*



# Suma de Impares

3

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      const int TAM = 15;
6
7      // Ingresar array de 15 posiciones
8      int arr[TAM];
9      cout << "Ingrese " << TAM << " numeros enteros:" << endl;
10     for (int i = 0; i < TAM; i++) {
11         cin >> arr[i];
12     }
13
14     // Calcular suma de elementos impares
15     int sumImpares = 0;
16
17     for (int i = 0; i < TAM; i++) {
18         if (arr[i] % 2 != 0) {
19             sumImpares += arr[i];
20         }
21     }
22
23     cout << "Suma de los valores impares: " << sumImpares << endl;
24
25     return 0;
26 }
27
```

Ingrese 15 numeros enteros:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

Suma de los valores impares: 64



Brian Cruceño 5to 5ta



# Explicacion del codigo

4

*#include <iostream>: Incluye la biblioteca estándar para entrada y salida, que permite usar cout para imprimir en la consola.*

*using namespace std;: Permite el uso de nombres estándar como cout sin necesidad de prefijarlos con std::.*

*int main() { :Inicia la función principal del programa. Es el punto de entrada.*

*cout << "Ingrese " << TAM << " numeros enteros:" << endl;*

```
for (int i = 0; i < TAM; i++){  
    cin >> arr[i];  
}
```

*Muestra un mensaje que solicita al usuario que ingrese 15 números enteros.*

*Un bucle for recorre las 15 posiciones del arreglo.*

*En cada iteración, se lee un número ingresado por el usuario y se almacena en el arreglo arr[i].*

Brian Cruceño 5to 5ta



# Explicacion del codigo

5

```
int sumImpares = 0;

for (int i = 0; i < TAM; i++){

    if (arr[i] % 2 != 0) {

        sumImpares += arr[i];

    }

}
```

Se declara la variable `sumImpares` e inicializa en 0, que almacenará la suma de los números impares.

Un bucle `for` recorre todo el arreglo. En cada iteración:

Se verifica si el número es impar usando la condición `arr[i] % 2 != 0`. Si es cierto, significa que el número es impar.

Si el número es impar, se suma a `sumImpares` usando la operación `sumImpares += arr[i]`.

`cout << "Suma de los valores impares: " << sumImpares << endl;`

Imprime el valor de `sumImpares`, que es la suma de todos los números impares en el arreglo.

`return 0;` Finaliza el programa devolviendo 0, lo que indica que la ejecución fue exitosa.



# Final del informe.

6



Materia: Laboratorio de Programación  
EEST N.o 5 - "Galileo Galilei" San Martín  
5° año 5° jueves de 7:40 a 11:55 Hs



*Brian Cruceño 5to 5ta*