



# **Maestría en Inteligencia Artificial Aplicada**

**Curso:** Cómputo en la nube (Gpo 10)

**Tecnológico de Monterrey**

**Alumno :** Fausto Velasco Atilano

**Profesor:** Dr. Eduardo Antonio  
Cendejas Castro

**Fecha :** 27 de Enero del 2023

**Actividad Modulo 2**

Programación de una solución paralela

# Introducción

La utilización de arreglos en la programación es muy común, por ejemplo piense en una suma de dos arreglos como la mostrada en la siguiente imagen, donde tenemos dos arreglos llamados A y B de 10 elementos cada uno y deseamos crear un tercer arreglo con la sumatoria de cada uno de los elementos en los mismos subíndices. En esta tarea se usa OpenMP desde visual Studio para acelerar este procesamiento

	Subíndices	0	1	2	3	4	5	6	7	8	9
Arreglo A		1	2	3	4	5	6	7	8	9	10
						+					
	Subíndices	0	1	2	3	4	5	6	7	8	9
Arreglo B		11	12	13	14	15	16	17	18	19	20
						=					
	Subíndices	0	1	2	3	4	5	6	7	8	9
Arreglo R		12	14	16	18	20	22	24	26	28	30






Este es el proceso que deseamos replicar, para lo que debes:

1. Crear un proyecto en Visual Studio
2. Configurarlos para que pueda utilizar la librería OpenMP
3. Crear los arreglos y asignarles valores aleatorios o pedirle los valores al usuario (la cantidad de elementos puede ser de 1000 para ejemplificar el proceso y obtener rápidamente resultados)
4. Realizar un for paralelo a través de las funciones de la librería OpenMP
5. Imprimir los arreglos o parte de ellos para comprobar que el arreglo resultante contiene la suma de los dos arreglos

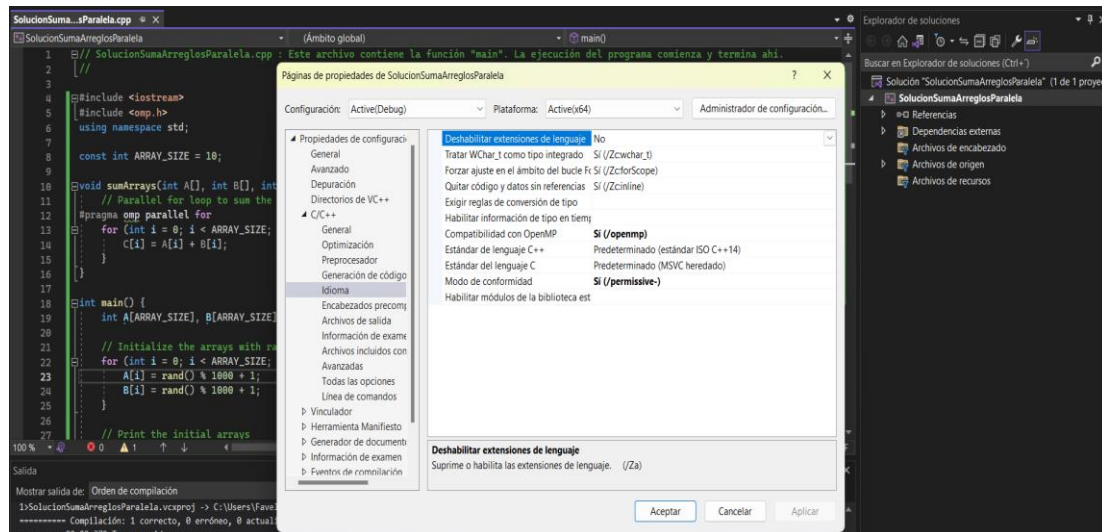
Al terminar deberá subir su proyecto a un repositorio personal de GitHub

# Liga de Repositorio & Capturas de pantalla

<https://github.com/Fausto9/ComputoenlanubeMNA>

Fausto9 files added			d4c549d 5 minutes ago	1 commit
	SolucionSumaArreglosParalela.cpp	files added	5 minutes ago	
	SolucionSumaArreglosParalela.sln	files added	5 minutes ago	
	SolucionSumaArreglosParalela.vcxproj	files added	5 minutes ago	
	SolucionSumaArreglosParalela.vcxpro...	files added	5 minutes ago	
	SolucionSumaArreglosParalela.vcxpro...	files added	5 minutes ago	

## Configuración del Proyecto en Visaul Studio



# Explicación del Código y Output

```
SolucionSuma...sParalela.cpp  X
SolucionSumaArreglosParalela  (Ámbito global)  main()
4  #include <iostream>
5  #include <omp.h>
6  using namespace std;
7
8  const int ARRAY_SIZE = 10;
9
10 void sumArrays(int A[], int B[], int C[]) {
11     // Parallel for loop to sum the elements of A and B
12     #pragma omp parallel for
13     for (int i = 0; i < ARRAY_SIZE; i++) {
14         C[i] = A[i] + B[i];
15     }
16 }
17
18 int main() {
19     int A[ARRAY_SIZE], B[ARRAY_SIZE], C[ARRAY_SIZE];
20
21     // Initialize the arrays with random values
22     for (int i = 0; i < ARRAY_SIZE; i++) {
23         A[i] = rand() % 1000 + 1;
24         B[i] = rand() % 1000 + 1;
25     }
26
27     // Print the initial arrays
28     cout << "Array A: ";
29     for (int i = 0; i < ARRAY_SIZE; i++) {
30         cout << A[i] << " ";
31     }
32     cout << endl;
33
34     cout << "Array B: ";
35     for (int i = 0; i < ARRAY_SIZE; i++) {
36         cout << B[i] << " ";
37     }
38     cout << endl;
39
40     // Sum the arrays
41     sumArrays(A, B, C);
```

El código define una función `fill_array` que llena un arreglo con números aleatorios entre 1 y 1000. Luego define una función `print_array` para imprimir los elementos del arreglo. La función `sum_arrays` suma los arreglos A y B y almacena el resultado en C. Se utiliza un `for` paralelo OpenMP con la directiva `#pragma omp parallel for` para acelerar el procesamiento de la suma de los arreglos.

*Output*

```
Consola de depuración de Mi  X  +  -
Array A: 42 335 170 479 963 706 282 962 996 828
Array B: 468 501 725 359 465 146 828 492 943 437
Array C (sums A and B): 510 836 895 838 1428 852 1110 1454 1939 1265

C:\Users\Favel\source\repos\SolucionSumaArreglosParalela\x64\Debug\SolucionSumaArreglosParalela.exe (proceso 61952) se c
erró con el código 0.
Presione cualquier tecla para cerrar esta ventana. . .|
```



# Conclusión y Reflexión

- La programación paralela es una técnica que permite dividir un problema en múltiples tareas que se ejecutan simultáneamente en diferentes núcleos o procesadores. Esto puede aumentar significativamente la velocidad de ejecución de un programa y mejorar el rendimiento en aplicaciones que realizan operaciones intensivas de cálculo.
- En el ejemplo anterior, se utiliza la librería OpenMP para acelerar la suma de arreglos mediante la creación de una tarea paralela para cada elemento del arreglo. Esto permite que varios elementos se sumen al mismo tiempo en lugar de uno por uno, lo que reduce el tiempo de ejecución del programa. Sin embargo, es importante notar que la programación paralela no es adecuada para todos los casos y puede tener un impacto negativo en el rendimiento si no se utiliza adecuadamente. Es necesario evaluar cuidadosamente si el problema es adecuado para la paralelización y elegir la mejor técnica para resolverlo.