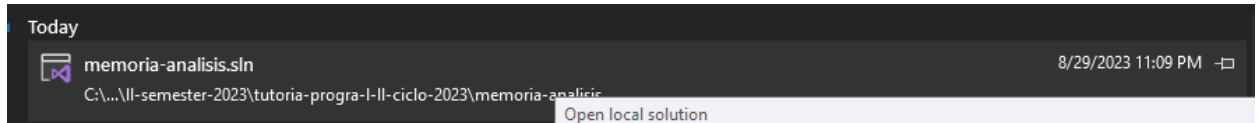


Tutora: Karla Verónica Quiros Delgado.

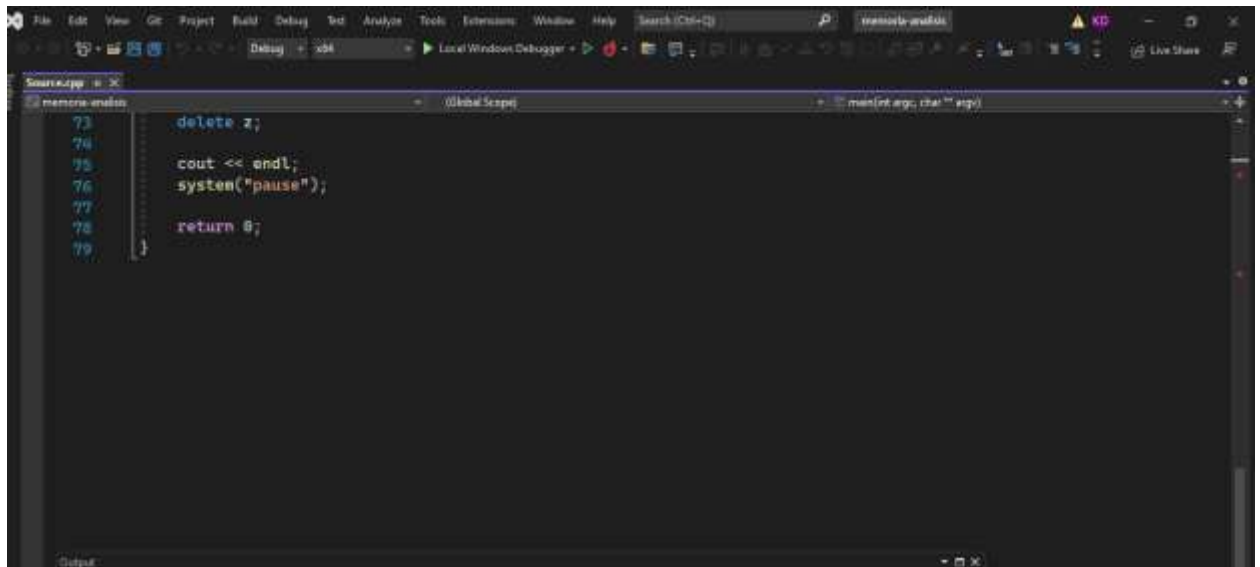
Fecha: 29-08-2023.

Debuggear básico

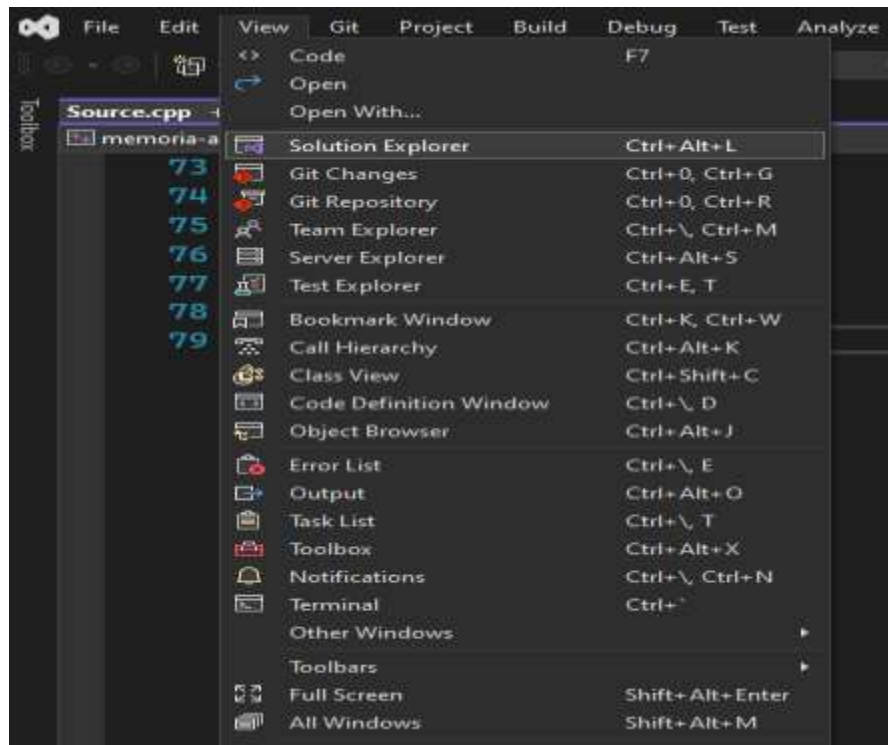
1. Abrir archivo en el que desea debuggear.



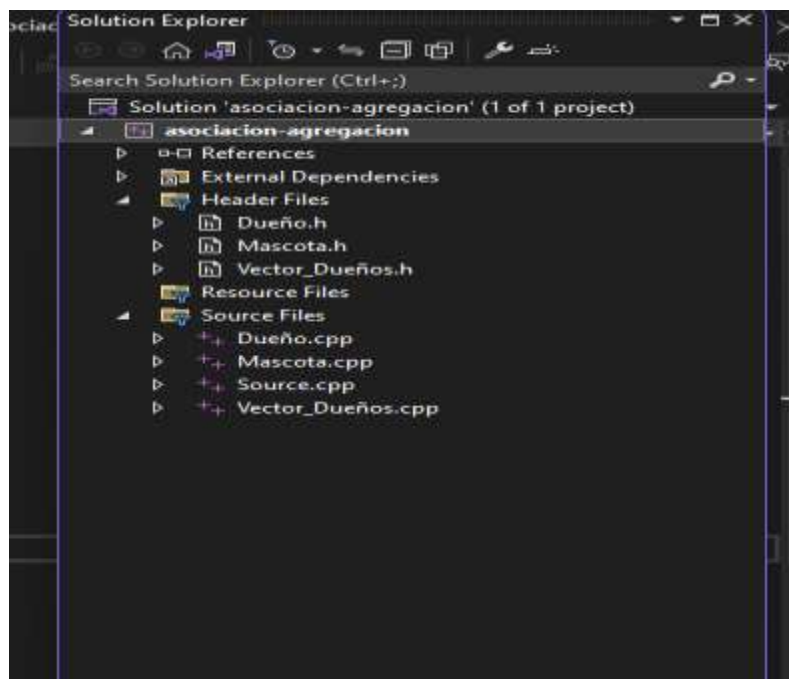
2. Si abre visual studio y no observa el explorador de archivos:



- **Ir a view (vistas)**



- Elegir explorador de soluciones y verán algo así:



Archivos cabecera y fuente del programa en cuestión.

3. Poner un punto de interrupción en la variable o función que desean debuggear:

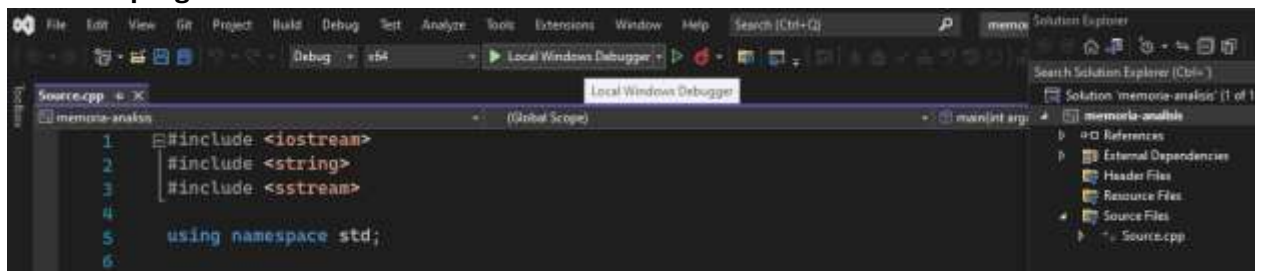
```

6
7   int main(int argc, char** argv) {
8       cout << "+++++ Ejercicio 1 +++++\n" << endl;
9
10      // Determine, antes de imprimir, la salida de cada instruccion.
11      int x = 10;
12      cout << " x          : " << x << endl; // imprime el valor de x
13      cout << " &x        : " << &x << endl; // imprime la direccion de mem
14      cout << " *(&x)      : " << *(&x) << endl; // (&x)=*, **= valor
15

```

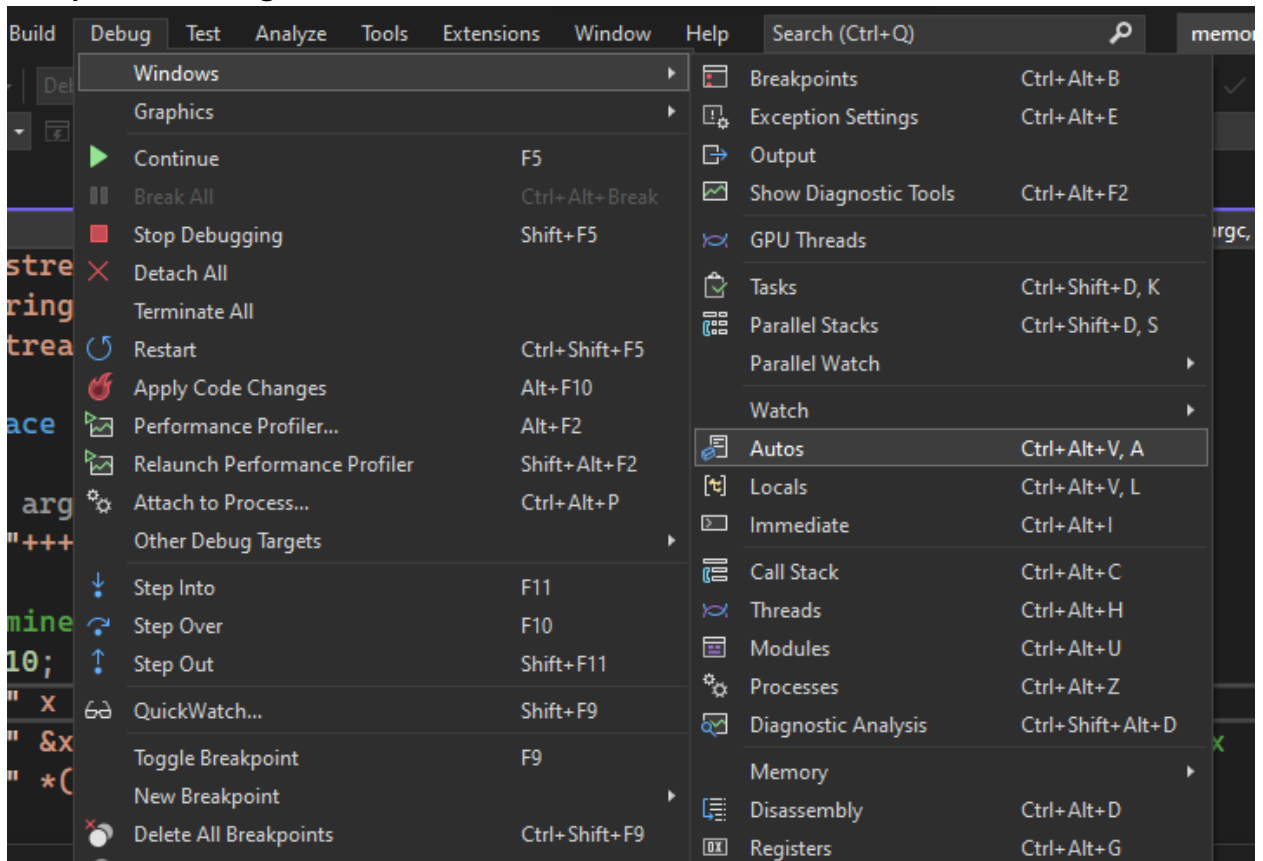
Se coloca en la barra gris del lado izquierdo antes de los números de línea.

4. Correr el programa



En la flecha verde.

5. Ir a opción de debug -> ventanas-> auto



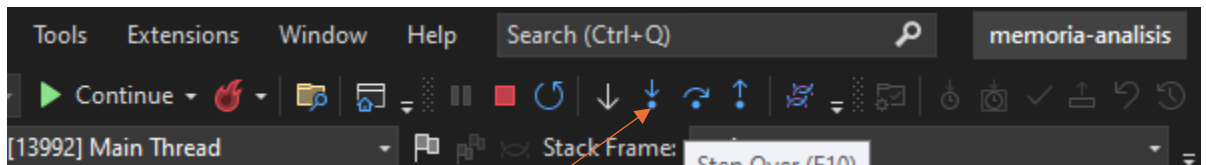
Aparecerá una barra con todas las variables del bloque en el que estamos debuggeando

```
1 #include <iostream>
2 #include <string>
3 #include <sstream>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     cout << "+++++ Ejercicio 1 +++++\n" << endl;
9
10    // Determine, antes de imprimir, la salida de cada instruccion.
11    int x = 10;
12    cout << " x          : " << x << endl; // imprime el valor de x
13    cout << " &x        : " << &x << endl; // imprime la direccion de memoria de x
14    cout << " *(&x)      : " << *(&x) << endl; // (*&x) == **&x, **&x = valor
15 }
```

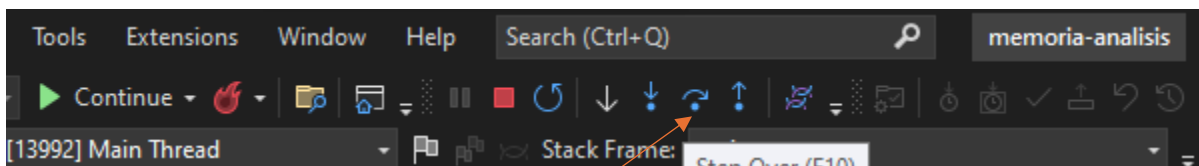
Autos:

Name	Value	Type
std::operator<< (std::char_traits<char> > returned)	{-}	std::basic_ostream< char, std::char_traits<char> >
x	0x000000b431ff74 (10)	int *
&x	0x000000b431ff88 (0x000000b431ff74 (10))	int *
*x	10	int
y	0x000000b431ff74 (10)	int *

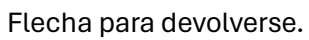
Donde observaremos como se comporta el programa y sus funciones y variables.



Flecha para entrar dentro de una función.



Flecha para saltar de línea.



Flecha para devolverse.