# Objetivos:

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

# Actividades:

* Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.
* Resolver un problema que requiera el uso de un arreglo de dos dimensiones, a través de un programa en lenguaje C.
* Manipular arreglos a través de índices y apuntadores.

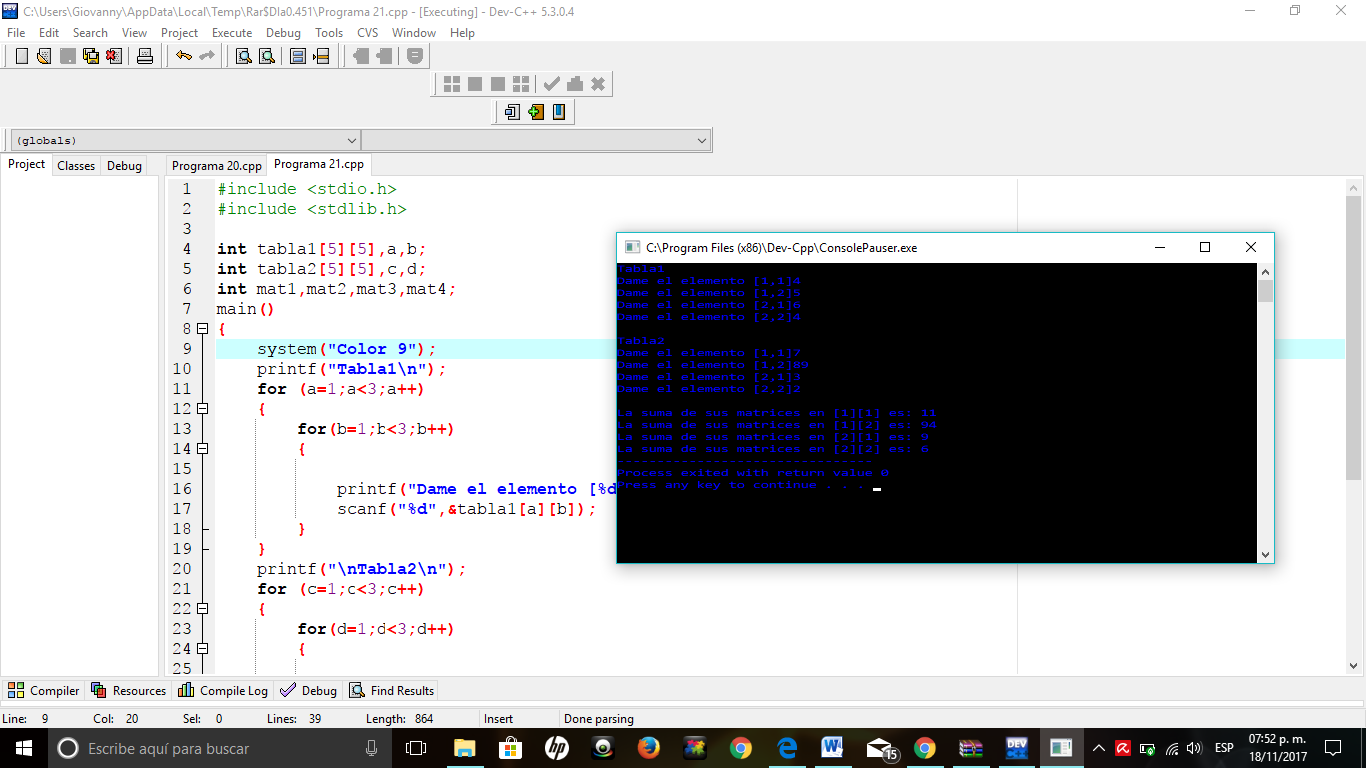
# Desarrollo;

Vimos una breve introducción sobre los arreglos, que son unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa. Vimos ejemplos del arreglo unidimensional con while y for.

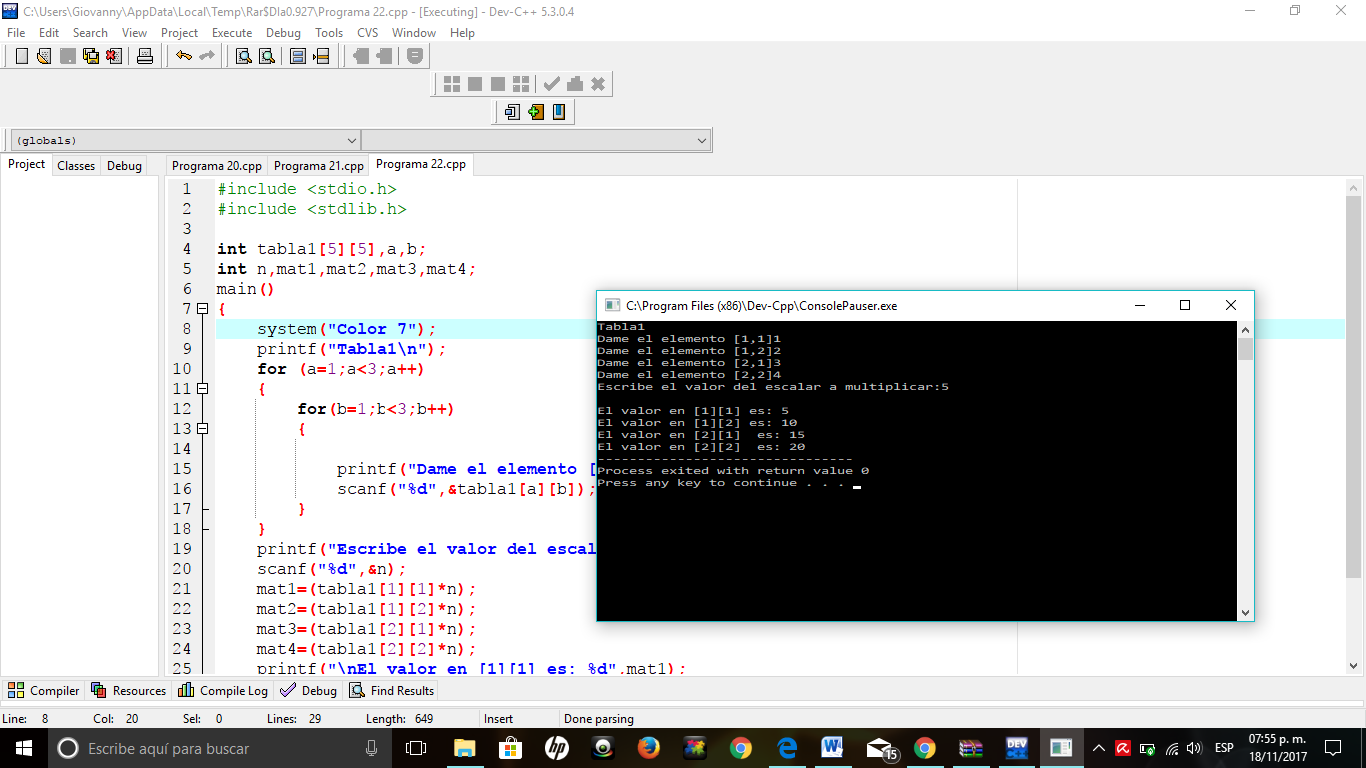
Pasamos a apuntadores, es una variable que contiene la dirección de una variable, es decir, hace referencia a la localidad de memoria de otra variable. Debido a que los apuntadores trabajan directamente con la memoria, a través de ellos se accede con rapidez a un dato. Y vimos varios ejemplos de ellos, con diferentes estructuras.

Continuamos con los arreglos multidimensionales, Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimensión. Los tipos de dato que puede tolerar un arreglo multidimensional son: entero, real, carácter o estructura. Y como en todos vimos ejemplos, esta práctica fue breve y un poco bien desarrollada.

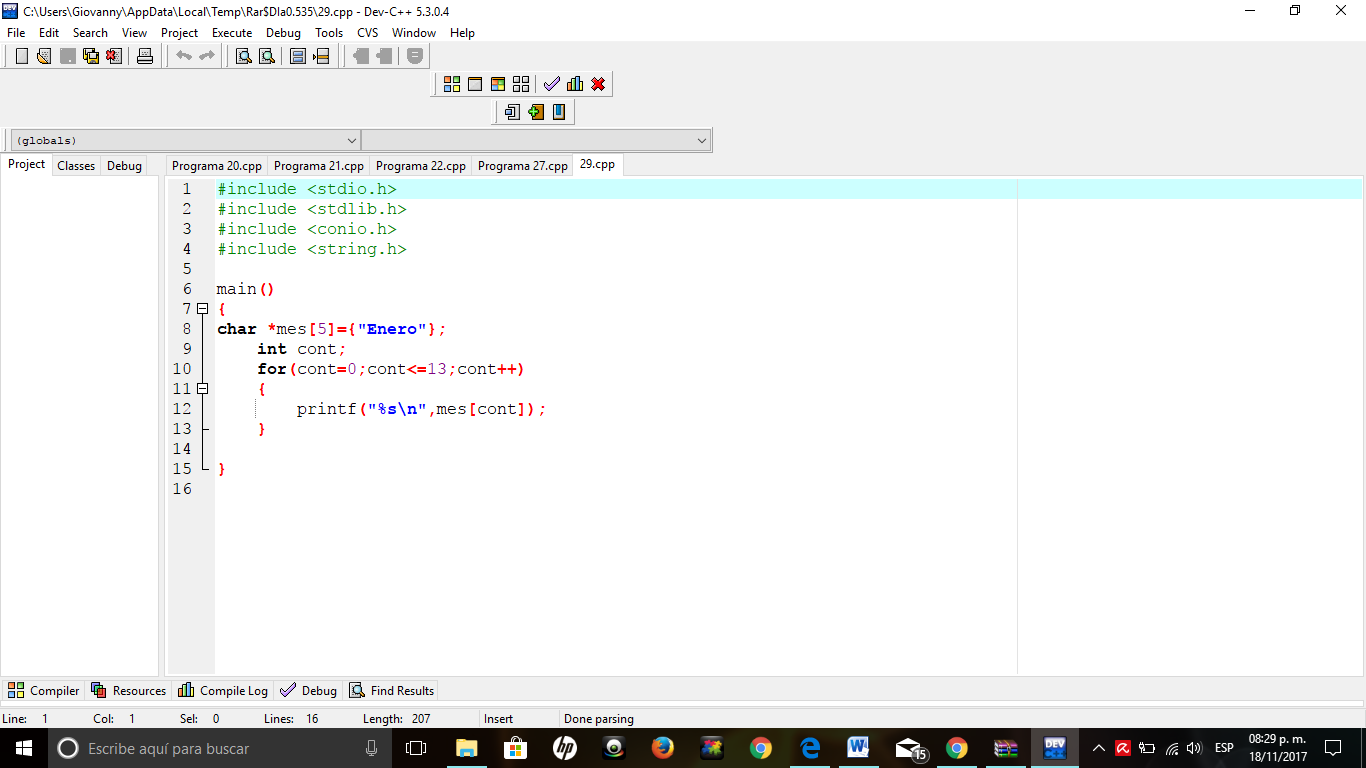
# Suma de matrices



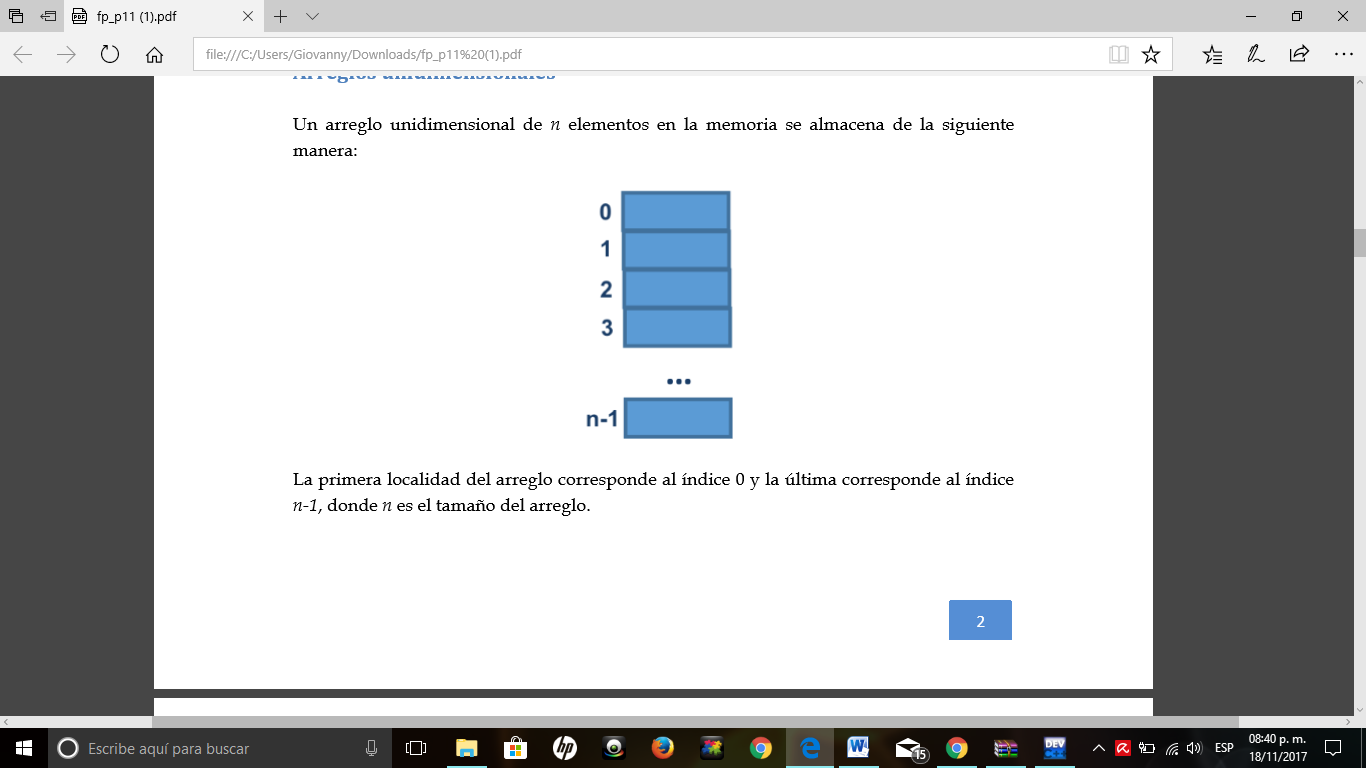
# Programa matriz por escalar



# Programa de los gastos anuales



# Conclusiones:

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse. A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices. Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.