**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**

****

**INTRODUCCIÓN A LA PROGRAMACIÓN**

**EQUIPO 1**

**338886 Omar Alonso Escápita Chacón**

**338852 Brian Jair Acosta Loera**

**338912 Juan Luis Del Valle Sotelo**

**338710 Mario Alberto Teran Acosta**

**Documentación del Tema de Exposición**

**Tercer Parcial**

**Arreglos de Dos o más Dimensiones**

**04 de noviembre de 2019**

**Tabla de contenido**

[Objetivo](#_ezljxthny2xg) **3**

[Desarrollo](#_6vxbh0djmnek) **3**

[¿Qué es un arreglo o array?](#_mkdn52pc4dbt) 3

[¿Para qué sirven los arreglos?](#_71nww0h9xijl) 4

[Arreglos Unidimensionales](#_t4a256bd2rba) 4

[Arreglos Bidimensionales](#_mmsiavytc63f) 5

[Arreglos Multidimensionales](#_wadf56rqj74i) 6

[Actividad de Aprendizaje](#_t4ssk6qurto9) **8**

[Aprendizajes](#_z4ax7r4g30e0) **9**

[Conclusiones](#_9oip6a3825v) **9**

[Recomendaciones](#_zakmiars9yjg) **10**

[Referencias](#_y6u303k6g7aa) **11**

# **Objetivo**

En este documento se presentará información acerca de los arreglos, ya sean unidimensionales, bidimensionales o con más de dos dimensiones, llamados multidimensionales. Se incluirá el uso de ejemplos para mostrar la sintaxis apropiada en C y una actividad de aprendizaje para que los alumnos puedan involucrarse en su uso y se familiaricen con ella al momento de realizar prácticas y futuros proyectos.

# **Desarrollo**

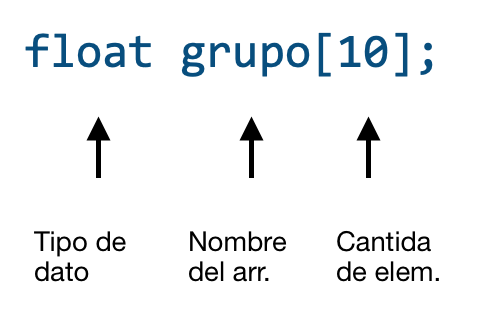
En el lenguaje C/C++ un arreglo puede tener varias dimensiones, por consiguiente hablaremos de los arreglos unidimensionales, bidimensionales y multidimensionales.

## **¿Qué es un arreglo o array?**

Los arreglos son estructuras de datos consistentes en un conjunto de datos del mismo tipo, ya sean enteros, caracteres, flotantes, entre otros. Estos contienen un tamaño, el cual es la cantidad de datos del mismo tipo que pueden almacenar.

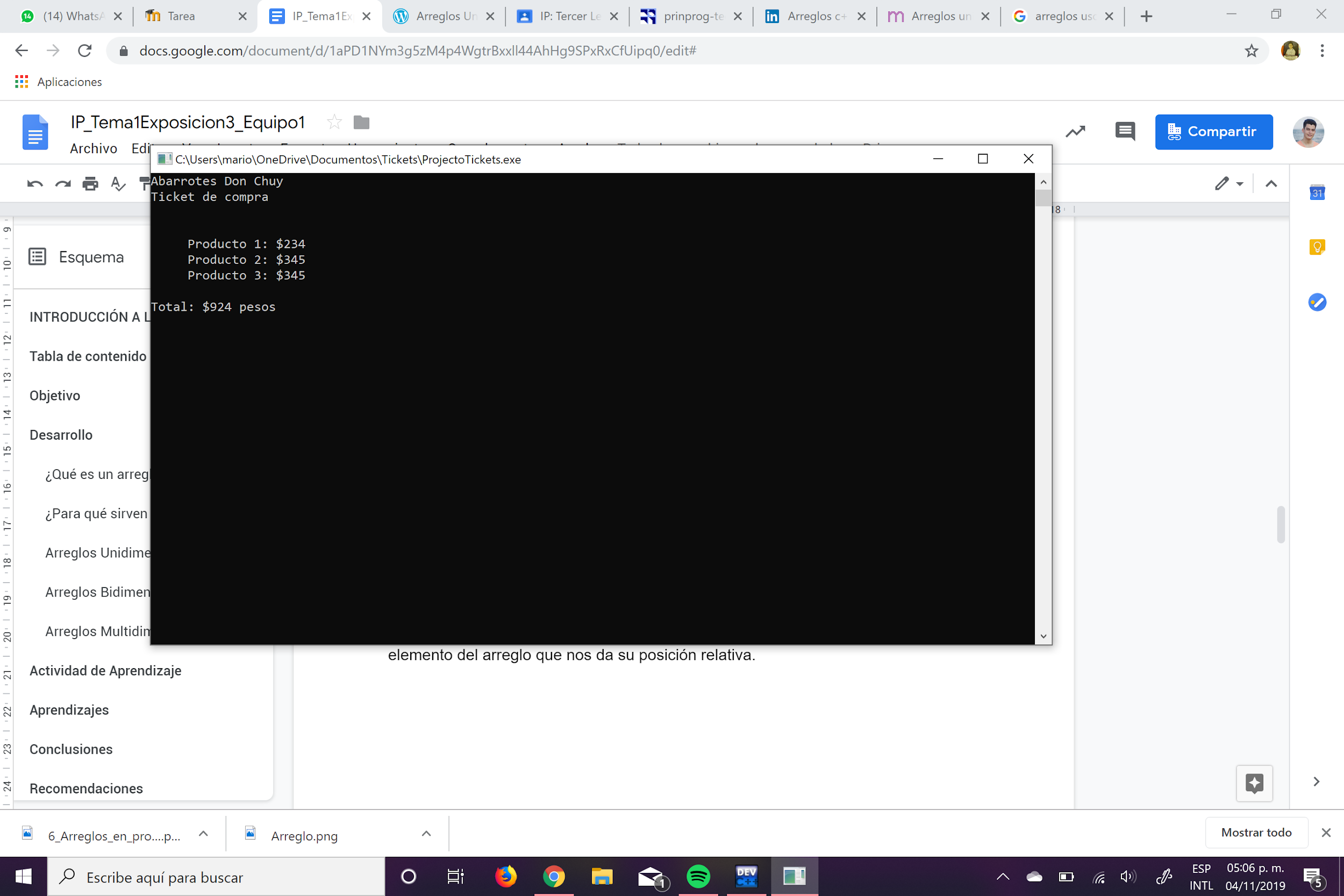
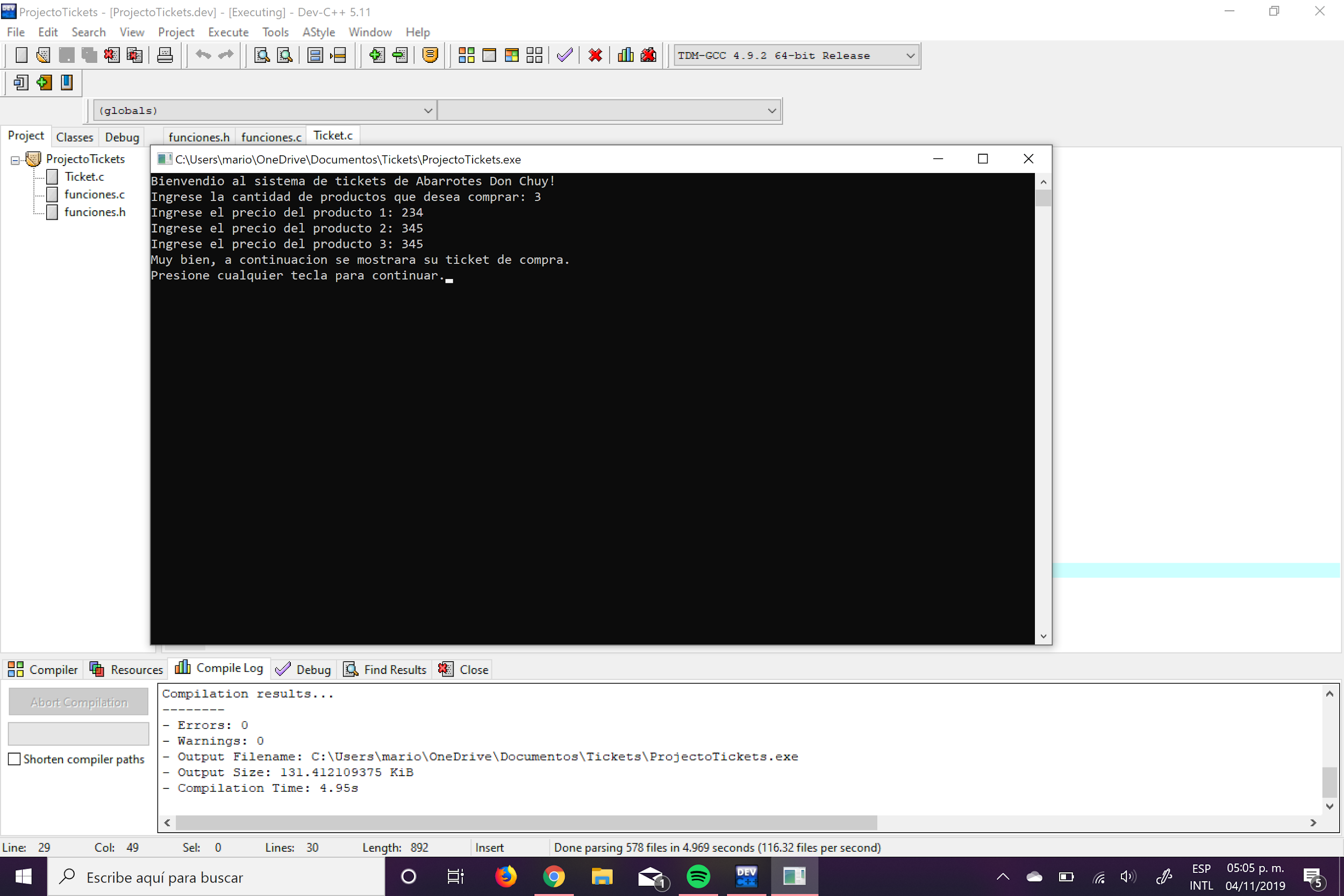
Los arreglos son entidades estáticas debido a que se declaran de un cierto tamaño y se mantienen con ese mismo tamaño durante toda la ejecución del programa. Los elementos de un arreglo de se pueden acceder mediante índices que indican del 0 hasta *n*-1, donde *n* es el tamaño del arreglo.

Su declaración consiste en el tipo de dato, seguido por el nombre del arreglo y su tamaño encerrado en corchetes [ ].



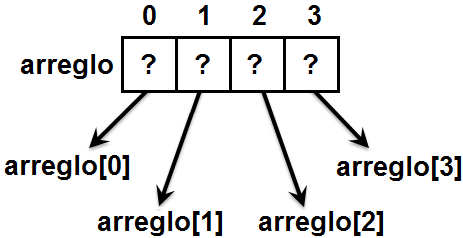
## **¿Para qué sirven los arreglos?**

Los arreglos nos permiten manejar de forma sencilla y directa conjuntos de datos del mismo tipo, de los cuales conocemos su cantidad de elementos y con los cuales se realizarán procedimientos y operaciones similares como ordenar, buscar, insertar, eliminar, modificar entre otras.



## **Arreglos Unidimensionales**

Un arreglo unidimensional consiste en una lista de datos del mismo tipo, guardados bajo el mismo nombre. Se caracterizan por ser un acceso directo, ya que podemos acceder a cualquier elemento del arreglo sin tener que consultar a elementos anteriores o posteriores, mediante el uso de un índice para cada elemento del arreglo que nos da su posición relativa.



La sintaxis de los arreglos unidimensionales es la siguiente:

tipoDeDato nombreDelArreglo [tamaño];

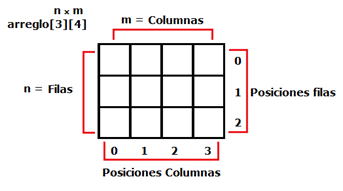
Donde “tipoDeDato” es el tipo de dato que queremos almacenar en el arreglo, ya sean enteros, caracteres, flotantes, entre otros. Por consiguiente, “nombreDelArreglo” se refiere al nombre que le vamos a asignar al arreglo y “tamaño”, el cual está encerrado entre corchetes, es donde introduciremos la cantidad de elementos que queremos almacenar.

Por consiguiente, le podemos asignar valores a cada uno de los elementos, escribiendo el elemento, seguido por un signo de igual y el valor que deseamos colocar, mientras este sea del mismo tipo de dato con el que inicializamos el arreglo.

nombreDelArreglo [indiceDelElemento] = valor;

## **Arreglos Bidimensionales**

Un arreglo de dos dimensiones puede representarse como una matriz o tabla, necesita de dos índices (fila y columna) para acceder a uno de sus elementos. Un arreglo de 2 dimensiones es usado para representar elementos en una tabla con columnas y filas, se puede acceder a cualquier elemento dentro del arreglo indicando el índice de la columna y la fila del arreglo.



La sintaxis de los arreglos bidimensionales es la siguiente:

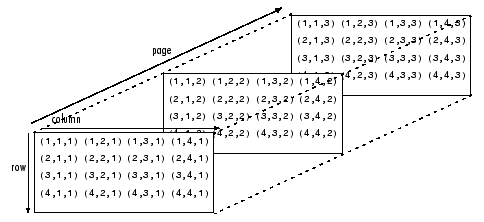
tipoDeDato nombreDelArreglo(matriz) [númeroDeFilas][númeroDeColumnas];

Donde “tipoDeDato” es el tipo de dato que queremos almacenar en el arreglo, ya sean enteros, caracteres, flotantes, entre otros. Por consiguiente, “nombreDelArreglo” se refiere al nombre que le vamos a asignar al arreglo y “tamaño”, el cual está encerrado entre corchetes, es donde introduciremos la cantidad de elementos que queremos almacenar.

nombreDelArreglo [indiceDelElementoX][indiceDelElementoY] = valor;

## **Arreglos Multidimensionales**

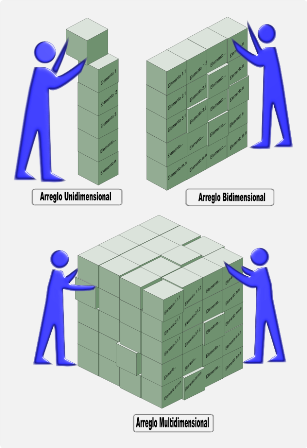
De igual forma que los arreglos unidimensionales, el almacenamiento de los datos en la memoria se realiza de forma secuencial y son accedidos mediante índices. Los arreglos multidimensionales son también conocidos como matrices. Cada elemento se define mediante dos subíndices, el índice de fila y el índice de columna. Las matrices multidimensionales son una extensión de matrices de dos dimensiones y utilizan subíndices adicionales para la indexación. Una matriz de tres dimensiones, por ejemplo, utiliza tres subsublos. Los dos primeros son igual que una matriz, pero la tercera dimensión representaría páginas en las que se encuentran ordenadas cada una de las matrices.



La sintaxis de los arreglos multidimensionales es idéntica a la de los bidimensionales:

tipoDeDato nombreDelArreglo(matriz)[dimensión1][dimensión2]...[dimensiónN]

Con N siendo un número entero positivo.



# Actividad de Aprendizaje

El alumno como prueba de su aprendizaje durante la exposición creará un programa en la aplicación de escritorio DEV C++, donde utilizara arreglos multidimensionales para resolver un problema sencillo algebraico.

El programa consiste en una suma de matrices, utilizando dos diferentes arreglos multidimensionales, donde el usuario proporcionará los elementos de la matriz y por consiguiente, realice la suma de estos.

#include <stdio.h>

void main(){

float a[100][100], b[100][100], c[100][100];

int i,j,m,n;

printf("Ingrese el numero de filas: ");

scanf("%i", &m);

printf("Ingrese el numero de columnas: ");

scanf("%i", &n);

printf("\nIntroduzca los datos de la matriz A:\n");

for(i=1; i<=m; i++)

{

for(j=1;j<=n;j++)

{

printf("\nIntroduzca la coordenada (%i,%i): ",i,j);

scanf("%f", &a[i][j]);

}

}

printf("\nIntroduzca los datos de la matriz B:\n");

for(i=1; i<=m; i++)

{

for(j=1;j<=n;j++)

{

printf("\nIntroduzca la coordenada (%i,%i): ",i,j);

scanf("%f", &b[i][j]);

}

}

for(i=1;i<=m;i++)

{

for(j=1;j<=n;j++)

c[i][j]= a[i][j] + b[i][j];

}

printf("\nEl resultado de la matriz C es:\n");

for(i=1;i<=m;i++)

{

for(j=1;j<=n;j++){

printf("(%i,%i): %10f\t", i,j,c[i][j]);

}

printf("\n");

}

}

# Aprendizajes

Durante esta investigación el equipo pudo comprender de mejor manera el funcionamiento de los arreglos y expandir el conocimiento a más áreas con los conceptos de arreglos de varias dimensiones y su estructura en el código, lo cual se pudo relacionar con el tema de matrices que estamos viendo en álgebra.

# **Conclusiones**

Una de las herramientas muy útiles en programación son los arreglos, los lenguajes de programación ofrecen la posibilidad de emplear matrices para trabajar y ordenar datos. Existen distintos tipos de arreglos para poder utilizar al realizar un proyecto, por lo que el tener conocimiento acerca de las variadas formas en las que estos recursos se pueden usar, así como la manera de manipularlos contribuye a la formación de los alumnos como futuros programadores.

# **Recomendaciones**

* Presentar casos en los que el uso de un arreglo multidimensional ayude a resolver un problema de una manera más eficiente.
* La práctica siempre ayuda al aprendizaje, asi que realizar varios ejercicios con arreglos ayudará a dominar el tema.

# 

# **Referencias**

Arreglos Unidimensionales y Multidimensionales. (2017, 8 marzo). Recuperado el 5

noviembre, 2019, de <https://clasesadsi.wordpress.com/2017/03/07/arreglos-unidimensionales-y-multidimensionales/>

Cisneros, E. (2009, 23 diciembre). Lenguaje C++ Arreglos unidimensionales y

multidimensionales. Recuperado el 4 noviembre, 2019, de

<https://aldeafraypedrodeagreda.files.wordpress.com/2009/12/11-arreglos-multidimensionales.pdf>

El Lenguaje C. (s.f.). El Lenguaje en C. Recuperado de la Internet el día 27 octubre, 2019, de

<https://www.fing.edu.uy/tecnoinf/mvd/cursos/prinprog/material/teo/prinprog-teorico07.pdf>

MathWorks. (s.f.). *Los arrays multidimensionales.* Recuperado el 4 de noviembre de 2019, de MathWorks: <https://la.mathworks.com/help/matlab/math/multidimensional-arrays.html>

Programación de computadores. (s. f.) Arreglos Bidimensionales. Recuperado en la Internet el día 4 de noviembre de 2019 en: <http://progra.usm.cl/apunte/materia/arreglos-2d.html>