

**Universidad Autónoma del Estado de México  
Centro Universitario Valle de Teotihuacán**



**Programa Educativo de la  
Licenciatura en Informática Administrativa**

**Manual de Prácticas de Estructura de Datos**

Elaboró: Dra. en T.I.E. Adriana Bustamante Almaraz

Fecha: 24/Enero/2023

Fecha de  
aprobación

\_\_\_\_\_  
H. Consejo Académico

\_\_\_\_\_  
H. Consejo de Gobierno

# INDICE

	PÁG.
PRESENTACIÓN.....	5
PRÁCTICA 1 .....	7
PRÁCTICA 2 .....	21
PRÁCTICA 3 .....	33
PRÁCTICA 4 .....	39
PRÁCTICA 5 .....	45
PRÁCTICA 6 .....	53
PRÁCTICA 7 .....	61
PRÁCTICA 8 .....	69
PRÁCTICA 9 .....	75
PRÁCTICA 10 .....	81



Universidad Autónoma del Estado de México

## PRESENTACIÓN

El presente manual de prácticas tiene por objeto guiar al alumno del programa educativo en Informática Administrativa con respecto a la unidad de aprendizaje Estructura de Datos en el cual aprenderá que es un entorno de desarrollo integrado, los conceptos básicos que conllevan la estructuración de datos bajo el lenguaje de programación de alto nivel y estructurado C++, conocerá los diferentes tipos de datos que se declaran en dicho lenguaje de programación, así como la creación de arreglos dinámicos, estáticos. Usará las diferentes instrucciones de decisión y cíclicas para manipular los diferentes tipos de estructuración de datos como el ordenamiento y búsqueda, los grafos y los árboles. Así mismo aprenderá a implementar los algoritmos de recursividad en esta unidad de aprendizaje.



## PRACTICA No. 1

### Mi Primer programa en C++

#### DURACIÓN:

Duración estimada para realizar la práctica: 60 min

#### INTRODUCCIÓN:

C++ es un lenguaje de programación diseñado en 1979, la intención de su creación fue extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

Para comenzar se debe incluir los siguientes comandos:

#### COMENTARIOS:

Los comentarios sirven para dejar plasmada una pequeña explicación de lo que se está haciendo en una línea o en un programa en total.

Los comentarios llevan pueden llevar dos formatos:

- \* El primero es para más de una línea, y su sintaxis es la siguiente:

```
/* ..... */
```

- \* El segundo es para solo una línea, y su sintaxis es la siguiente:

```
// .....
```

//: Esta instrucción nos indica que la línea es un comentario, esto nos puede ayudar a saber o recordar lo que escribimos, al momento de ejecutar el programa la computadora lo ve y lo ignora, si es que el comentario tiene muchas líneas también se puede usar esta instrucción que empieza con `/*` y termina con `*/`.



## ENCABEZADOS:

**#:** El símbolo de gato es el que indica el uso de una directiva del procesador.

### **#include <iostream.h>**

Contiene las funciones de ingresar y mostrar datos.

### **#include <math.h>**

Contiene las funciones matemáticas comunes.

### **#include <time.h>**

Contiene las funciones para tratamiento y conversión entre formatos de fecha y hora.

### **#include <stdlib.h>**

Contiene tipos, macros y funciones para la conversión numérica, generación de números aleatorios, búsquedas y ordenación, gestión de memoria y tareas similares.

### **#include <string.h>**

Contiene los prototipos de las funciones y macros de clasificación de caracteres

### **#include <ctype.h>**

Contiene varias funciones para comprobación de tipos y transformación de caracteres

### **#include <conio.h>**

Contiene los prototipos de las funciones, macros, y constantes para preparar y manipular la consola en modo texto en el entorno de MS-DOS

## SIGNOS DE PUNTUACIÓN Y SEPARADORES

Todas las sentencias deben terminar escritas con un punto y coma. Otros signos de puntuación son:

! % ^ & \* ( ) - + = { } [ ] \ ; ' : < > ? , . / " "

Cada instrucción en C++ debe terminar con un **punto y coma** (también conocido como terminador de instrucciones)



**() Entre Paréntesis:**

Agrupar expresiones; alterar la precedencia normal de los operadores y su asociatividad; aislar expresiones condicionales; indicar llamadas a funciones, y señalar los parámetros de estas.

**Llaves { }:**

Señalan el comienzo y final de una sentencia compuesta, como los bloques de código.

**Coma,:**

La coma como puntuador se utiliza para separar los elementos en las listas de parámetros de una función.

**Dos puntos:**

Los dos puntos se utilizan para señalar sentencias etiquetadas.

**Int main:**

Esta instrucción funciona para dar inicio a un programa solo se declara una vez dentro del programa, todas las instrucciones del programa principal se deben escribir entre las llaves de apertura y cierre “{}”.

**Punto y coma,;**

El punto y coma va al final de cada sentencia indicando el final de la instrucción.

**Std::cout <<" ";**

Indica a la computadora que debe realizar la acción de imprimir los caracteres entre las comillas dobles.

SECUENCIAS DE ESCAPE COMUNES:

En c++ se usan símbolos para indicar alguna función específica, por ejemplo, al colocar \n le indicamos al programa que debe dar un “entre o salto de línea”, por ejemplo:

```
cout<<" \n Programando \n";
```

Indicamos que antes de que muestre la palabra “programando” debe haber un espacio vacío como si se hubiese dado un enter, y así mismo después de la palabra “programando” debe haber otro espacio en blanco o salto de línea.



**\n Nueva línea:**

Coloca el cursor de la pantalla al inicio de la siguiente línea.

**\t Tabulador horizontal:**

Desplaza el cursor de la pantalla hasta la siguiente posición de tabulación.

**\r Retorno de carro:**

Coloca el cursor de la pantalla al inicio de la línea actual; no avanza a la siguiente línea.

**\a Alerta:**

Suena la campana del sistema.

**\\ Barra diagonal inversa:**

Se usa para imprimir un carácter de barra diagonal inversa.

**\' Comilla sencilla:**

Se usa para imprimir un carácter de comilla sencilla.

**\" Doble comilla:**

Se usa para imprimir un carácter de doble comilla.

SECCIONES QUE INTEGRAN EL CUERPO DEL PROGRAMA EN C++:

```
1- /******. *****/
2
3 /* UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO */
4 /* CENTRO UNIVERSITARIO VALLE DE TEOIHUACAN */
5 /* LICENCIATURA EN INFORMATICA ADMINISTRATIVA 2º SEM */
6 /* UNIDAD DE APRENDIZAJE: ESTRUCTURA DE DATOS */
7 /* NOMBRE DEL ALUMNO: */
8 /* DOCENTE: Dra. en TIE. ADRIANA BUSTAMANTE ALMARAZ */
9 /* PRACTICA 1 */
10 /* FECHA DE ENTREGA: | */
11
12 *****/
13
14 #include <iostream>
15
16 using namespace std;
17
18 int main()
19 {
20     void PROGRAMA_UNO()
21     {
22         system("cls");
23         std::cout<<"Hola\n";
24         std::cout<<"Soy un excelente estudiante y este es ";
25         std::cout<<"Mi primer programa \n";
26     }
27 }
28
29
30 return 0;
31 }
```



## OBJETIVO(S):

Identificar las secciones que integran la estructura de un código en el lenguaje de programación C++.

Imprimir un mensaje al usuario mediante las instrucciones de la librería estandar de entrada y salida en el lenguaje de programación C++.

## REQUISITOS:

Elaborar el programa analizado en clases anteriores en el IDE Dev C++ que contiene el compilador de C++ bajo las siguientes características:

1. Realizar un encabezado de datos institucionales, personales y de la unidad de aprendizaje en modo comentario /\* \*/
2. Posteriormente capturar el código en c++ en el orden mostrado en clase
3. Realizar un reporte de práctica en el formato oficial en word adjunto en esta asignación con el contenido siguiente:
  1. **PORTADA.** Con datos de identificación: Nombre de la Universidad y del Centro Universitario, Nombre del alumno, nombre de la UA, Número y nombre de la práctica, Licenciatura, semestre, Docente: Dra. en T.I.E. Adriana Bustamante Almaraz.
  2. En la primera sección de **INTRODUCCIÓN** como parte del marco teórico, investiga los conceptos teóricos de las instrucciones vistas en clase para reforzar la inducción teórica de los mismos, en este caso señala cada una de las partes de tu programa y define para que sirve cada una de estas secciones.
  3. Para la sección de **DESARROLLO** de la práctica Obtén las capturas de pantalla del código previamente capturado en DEV C++, la ejecución del programa y el resultado deseado después de la ejecución.
  4. Al finalizar coloca una sección de **CONCLUSIÓN PERSONAL** que contenga una redacción de tu experiencia con el uso y manejo de DEV C++ describe lo que aprendiste, la experiencia que tuviste, ¿Que fue lo más fácil y qué lo mas difícil que comprendiste de estos temas?
  5. En esta ultima sección de **RECURSOS EN LÍNEA** coloca los links proporcionados para el desarrollo de esta práctica, así mismo si utilizaste otros recursos digitales colócalos en esta sección.

La práctica puede ser entregada en formato word o PDF siempre y cuando cubra los requerimientos arriba mencionados.



Universidad Autónoma del Estado de México

**NOTA IMPORTANTE:** A partir de esta práctica se entrega el archivo \*.CPP que genera Dev C++ con el código desarrollado en el mismo



### MATERIAL Y EQUIPO A UTILIZAR:

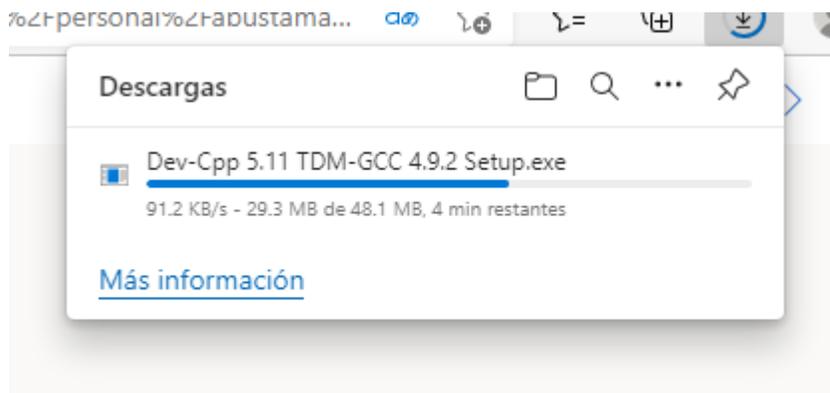
El material y equipo a utilizar para realizar la práctica es el siguiente:

- PC con IDE Dev C++ versión 5.11 instalado
- Manual de prácticas
- Bibliografía recomendada en este manual

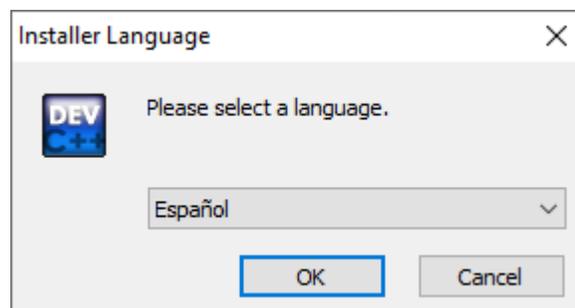


### DESARROLLO:

1. Descargar el IDE Dev-C++ 5.11 desde el sitio oficial de bloodshed:  
<https://www.bloodshed.net/>



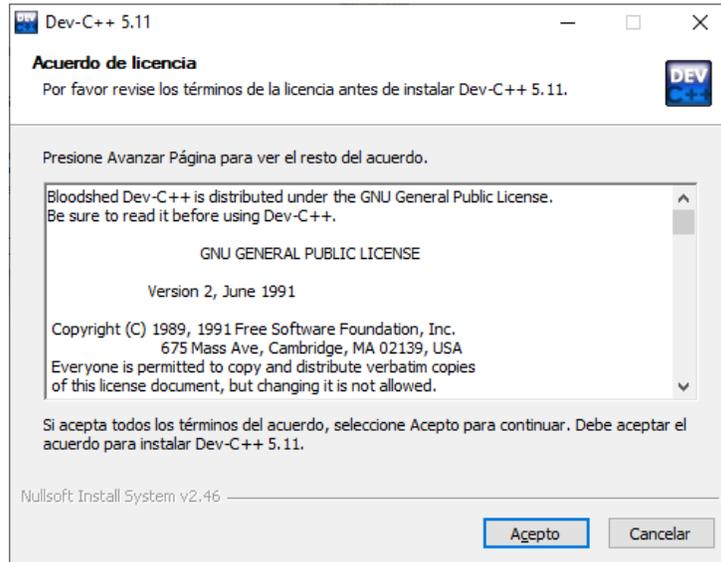
2. Una vez descargado se procede a instalar el IDE Dev-C++ 5.11, selecciona el idioma que aparecerá en el proceso de instalación



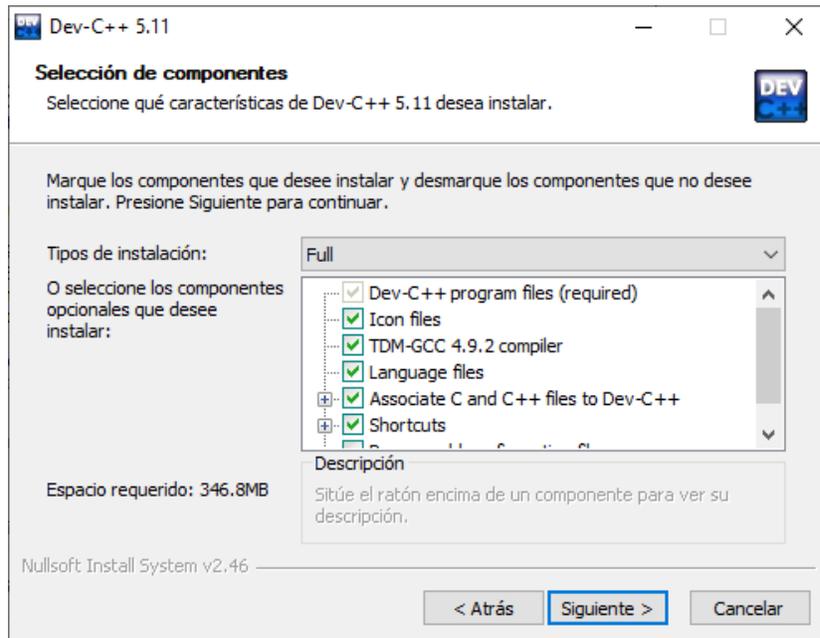


Universidad Autónoma del Estado de México

3. Posteriormente aceptar los acuerdos de licencia, se procede a la instalación



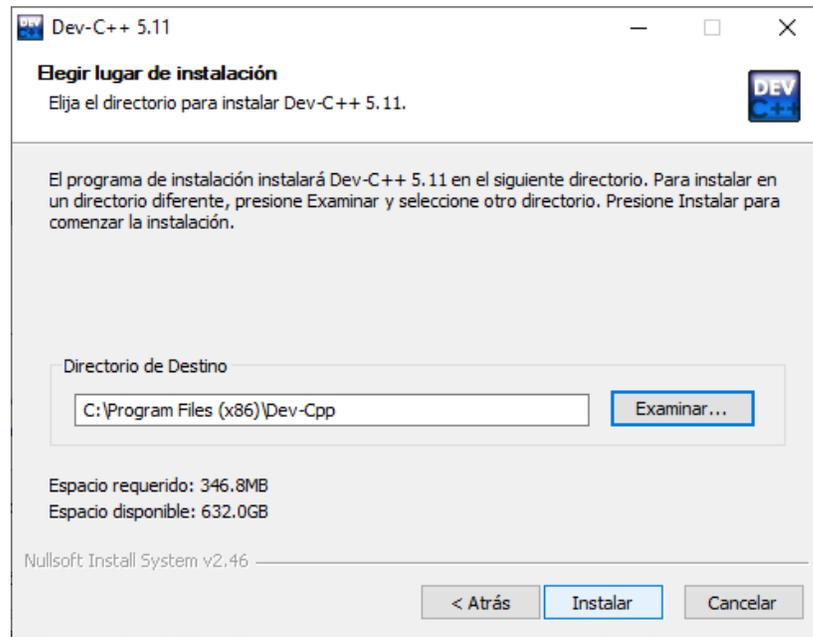
4. A continuación se mostrarán los componentes “las características que desea instalar” seleccionar las acciones que desea que realice Dev-C++ 5.11



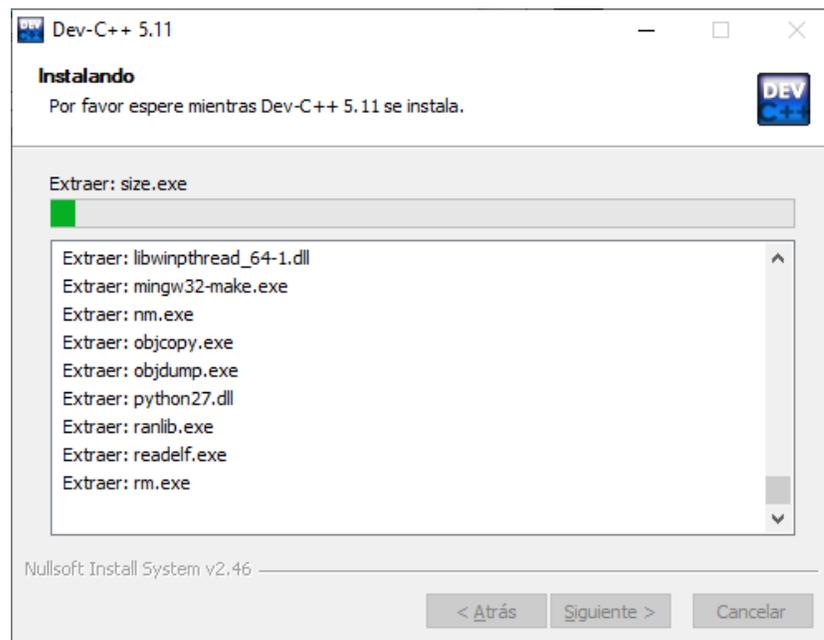


Universidad Autónoma del Estado de México

5. Seleccionar la carpeta en que desea guardar los datos de instalación o bien utilizar la que se sugiere por defecto



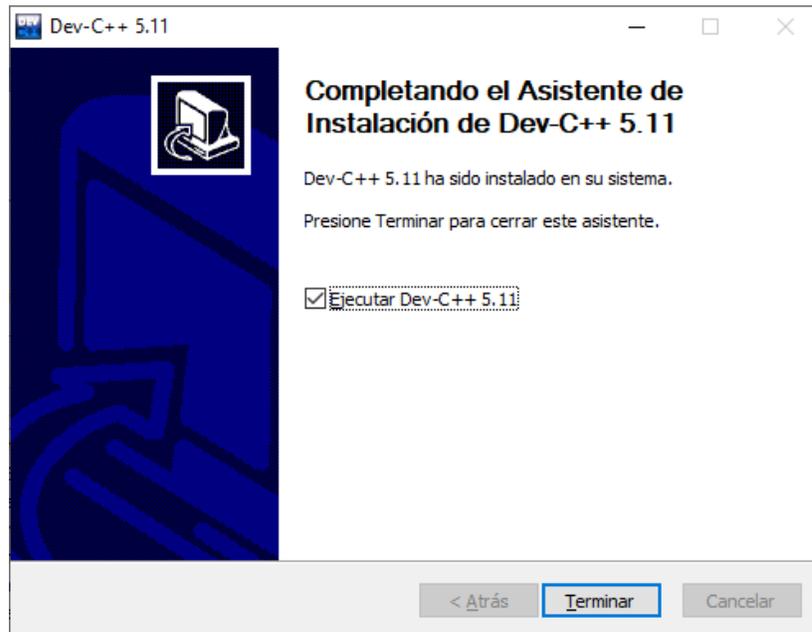
6. El setup procede a instalar el IDE Dev-C++ 5.11, esperar unos minutos a que culmine la instalación



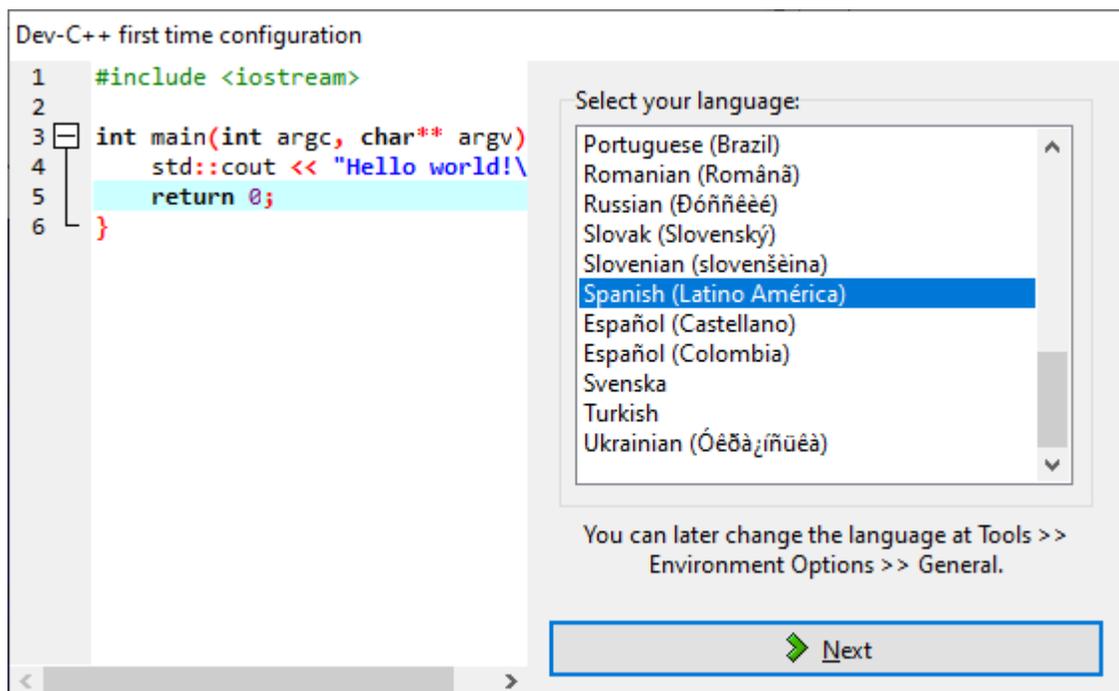


Universidad Autónoma del Estado de México

7. Finalmente culmina la instalación, al hacer clic en el boton “Terminar” se ejecutará el IDE Dev-C++ 5.11 por primera vez



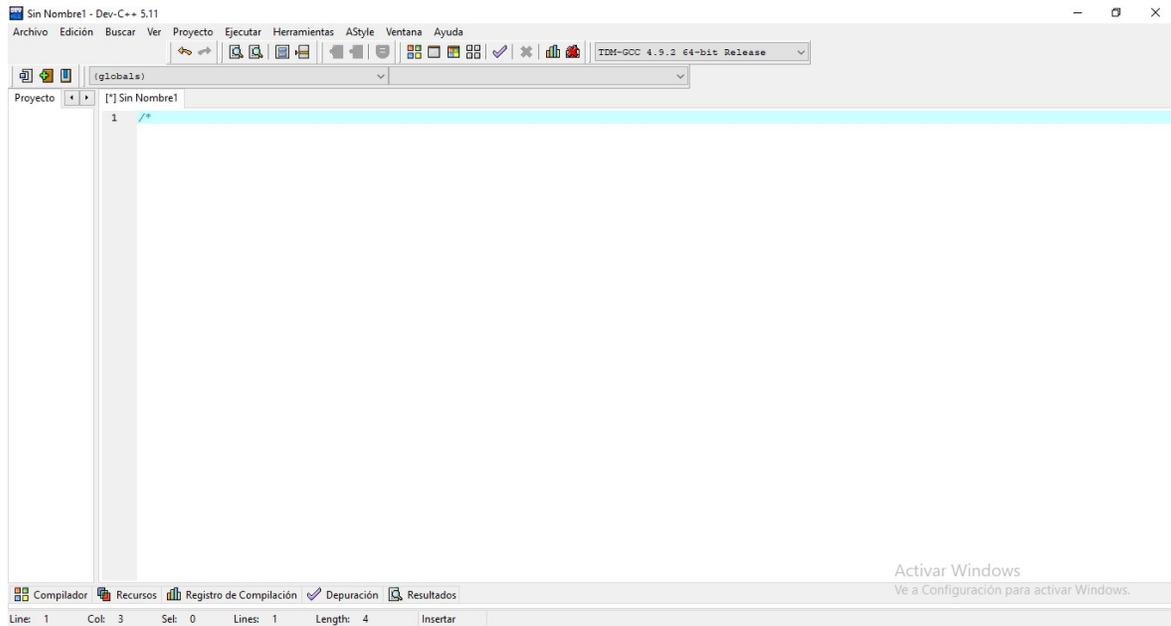
8. Una vez inicializado IDE Dev-C++ 5.11 solicita algunas características de configuración y personalización del entorno gráfico para el programador tales como: el idioma, color de fondo, tipo de letra e iconos. En caso de que no se desee personalización alguna es posible trabajar con la versión de la interfáz gráfica de usuario por defecto.



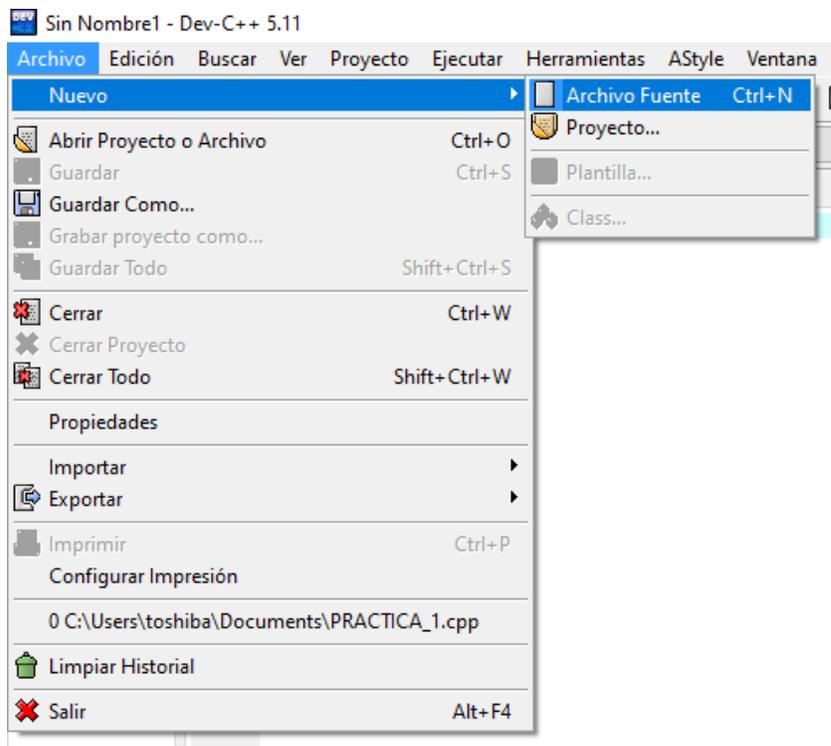


Universidad Autónoma del Estado de México

9. El editor IDE Dev-C++ 5.11 esta listo para escribir código fuente en el lenguaje de programación C++



10. Para iniciar la escritura de código fuente en un espacio de trabajo en blanco hacer clic en el menú Archivo -> Nuevo -> Archivo Fuente o utilizar el atajo rápido presionando las teclas: Ctrl+N





11. Escribir el código fuente con base en lo señalado en la sección de requisitos de este manual así como en lo visto en clase. Para realizar anotaciones en forma de comentario sin que el compilador de C++ lo identifique en la ejecución como instrucciones se colocan los caracteres `/* */` para colocar las anotaciones necesarias dentro del programa a manera de comentarios al inicio y al final o bien también se usan los caracteres `//` al inicio del texto. En esta sección del encabezado del programa coloca tus datos de identificación.

```
F:\Practica_1.cpp - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
[*] Practica_1.cpp.cpp
1  /*UNIVERSIDAD AUTÓNOMA DE ESTADO DE MÉXICO*/
2  /*CENTRO UNIVERSITARIO DEL VALE DE TEOTHUACÁN*/
3  /*LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA*/
4  /*2º SEMESTRE*/
5  /*UNIDAD DE APRENDIZAJE: ESTRUCTURA DE DATOS*/
6  /*NOMBRE DEL ALUMNO: JOSE ANTONIO SAMPERIO HERRERA*/
7  /*DOCENTE: DRA EN T.I.C. ADRIANA BUSTAMANTE ALNARAZ*/
8  /*PRACTICA 1. MI PRIMER PROGRAMA EN C++*/
9
Compilador Recursos Registro de Compilación Depuración Resultados
Line: 9 Col: 1 Sel: 0 Lines: 9 Length: 364 Insertar Done parsing in 0.031 seconds
```

12. Una vez concluida la sección de comentarios coloca la segunda parte del encabezado del programa que es la sección de librerías, en este caso se hará uso de: `#include <iostream>` la cual contiene instrucciones entrada y salida de datos, además de que se debe escribir el identificador `using` para indicarle al programa que se estarán utilizando dichas instrucciones de entrada y salida

```
13  #include <iostream>
14
15  using namespace std;
```



13. Después se procede a teclear la función principal `main ( )` ya que indica el inicio principal del programa así como las llaves de apertura y cierre `{ }` dentro de estas se escriben las instrucciones a ejecutarse por el programa

```
12
13     main()
14     {
15
```

14. En esa sección se escriben las instrucciones de salida que ejecutará el programa, unicamente se mandarán mensajes en pantalla iniciando con la instrucción de la librería `iostream: std::` que indica imprimir en pantalla `Cout<<` todo el texto escrito dentro de este. El texto debe ser escrito entre `""` para visualizarlo. Para proporcionar un formato al texto se utilizan las secuencias de escape, en este caso se colocará: `\n` al inicio y final del mensaje escrito. Al finalizar la escritura de cada línea de texto despues de la comilla(“) de cierre colocar el carácter `(;)` punto y coma, ya que indica que ha culminado la escritura de esa línea.

```
15
16     std::cout<<"\n ***BIENVENIDO AL SEMESTRE 2022_B***\n";
17     std::cout<<"\n HOLA YO SOY JOSE ANTONIO SAMPERIO HERRERA, TENGO 18 AÑOS, VIVO EN NOPATEPEC Y ESTOY ESTUDIANDO UNA LICENCIATURA EN INFORMATICA ADMINISTRATIVA, ME GUSTA M
18     std::cout<<"\n ¿TE GUSTA LA PROGRAMACION?";
19
```

15. Para la sección de finalización en el cuerpo del programa se coloca la instrucción `return 0;` para que en caso de una compilación y ejecución exitosa del programa, el compilador del lenguaje C++ devuelve el valor 0 al sistema operativo, regresando el control de funcionamiento y operación correcta del mismo. Posterior a la instrucción `return 0;` se escribe en la siguiente linea el carácter llave de cierre `}` para indicar que el código ha culminado de escribirse.



## Universidad Autónoma del Estado de México

```
1 /*UNIVERSIDAD AUTÓNOMA DE ESTADO DE MÉXICO*/
2 /*CENTRO UNIVERSITARIO DEL VALE DE TEOTIHUACÁN*/
3 /*LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA*/
4 /*2º SEMESTRE*/
5 /*UNIDAD DE APRENDIZAJE: ESTRUCTURA DE DATOS*/
6 /*NOMBRE DEL ALUMNO: JOSE ANTONIO SAMPERIO HERRERA*/
7 /*FECHA: 22/02/2022
8 /*DOCENTE: DRA EN T.I.C. ADRIANA BUSTAMANTE ALMARAZ*/
9 /*PRACTICA 1. MI PRIMER PROGRAMA EN C++*/
10
11 #include <iostream>
12
13 main()
14 {
15
16     std::cout<<"\n ***BIENVENIDO AL SEMESTRE 2022_B***\n";
17     std::cout<<"\n HOLA YO SOY JOSE ANTONIO SAMPERIO HERRERA, TENGO 18 AÑOS, VIVO EN NOPATEPEC Y ESTOY ESTUDIANDO UNA LICENCIATURA EN INFORMATICA ADMINISTRATIVA, ME GUSTA P
18     std::cout<<"\n ¿TE GUSTA LA PROGRAMACION?";
19
20     return 0;
21 }
22
```

16. Una vez terminada la escritura del código en el editor del IDE Dev-C++ 5.11 hacer clic en el menú Ejecutar -> Compilar (o presionando la combinación de teclas: Ctrl+F9) para verificar que no haya ningún error en la escritura del código de programación

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: F:\Practica_1.exe
- Output Size: 1.83291149139404 MiB
- Compilation Time: 1.81s
```

17. Una vez que no se generó ningún error u observaciones por parte del compilador se ejecuta el programa y se muestran los resultados deseados por el programador.

```
***BIENVENIDO AL SEMESTRE 2022_B***

HOLA YO SOY JOSE ANTONIO SAMPERIO HERRERA, TENGO 18 ADOS, VIVO EN NOPATEPEC Y ESTOY ESTUDIANDO UNA LICENCIATURA EN INF
ORMATICA ADMINISTRATIVA, VIVO CON MIS PAPA

¿TE GUSTA LA PROGRAMACION?

Process exited after 0.1304 seconds with return value 0
Presione una tecla para continuar . . .
```



## CONCLUSIONES:

En esta sección el alumno aprende a instalar el entorno de desarrollo integrado Dev C++ versión 5.11, comprende las secciones de un programa en lenguaje C++ a través de la escritura de un programa con salida de datos.

## CUESTIONARIO:

1. Con estas directivas del preprocesador se mandan a llamar las librerías de lenguaje C++ las cuales contienen las instrucciones clasificadas dentro de estas para escribir un código.
  - a) `#include <stdlib.h>, #include <conio.h>`
  - b) `#include <iostream>, #define`
  - c) `#define, const, void`
  - d) Ninguna de las anteriores
2. Con la siguiente línea de código en C++ se imprime un mensaje de salida al usuario en pantalla una vez compilado el programa correctamente:
  - a) `cin>>usuario;`
  - b) `cin.getline (usuario, 40,'\n');`
  - c) `cout <<"dame el primer número:";`
  - d) `cout<<"Introduce tu nombre de usuario:";`
3. Es la instrucción que permite al compilador devolver un valor al sistema operativo donde se ejecutó el programa para indicar que el proceso de compilación del código en C++ culminó correctamente:
  - a) `Return n;`
  - b) `return 0;`
  - c) `VOID PROYECTO( );`
  - d) `Int main ( )`
4. Es un programa informático que traduce un programa escrito en un lenguaje de programación, definido código fuente, a otro lenguaje de alto nivel:



Universidad Autónoma del Estado de México

- a) Código
  - b) Lenguaje de programación
  - c) Compilador
  - d) Interprete
5. Conjunto de órdenes desarrolladas estructuradamente para que una computadora realice un conjunto de tareas determinadas:
- a) Lenguaje máquina
  - b) Lenguaje de programación
  - c) Programación
  - d) Programa



#### REFERENCIAS

Bustamante Almaraz, A. (09 de Febrero de 2021). *Microsoft stream canal personal de Adriana Bustamante Almaraz*. Obtenido de Mi primer programa en C++:  
<https://web.microsoftstream.com/video/26d407db-6617-413e-bbab-ee524ee5a731?classId=05f2fefc-7678-4374-b4eb-6767b168d858&assignmentId=52a91ecd-a5e5-4a11-9646-bf2350cad999>



## PRACTICA No. 2

### Tipo de cambio monedas extranjeras y color de texto

#### DURACIÓN:

Duración estimada para realizar la práctica: 60 min

#### INTRODUCCIÓN:

¿Qué es una directiva de preprocesador?

Todas las directivas del preprocesador empiezan con #; y solo pueden aparecer espacios en blanco antes de una directiva del preprocesador en una línea. Las directivas del preprocesador se procesan por completo antes de que empiece la compilación.

#### Directiva #define:

La directiva del preprocesador #define crea constantes simbólicas (constantes representadas como símbolos) y macros (operaciones definidas como símbolos). Esta sección se incluye para beneficio de los programadores de C++ que necesiten trabajar con código heredado de C. En C++, las macros se pueden reemplazar comúnmente por las plantillas y funciones en línea. Una macro es una operación definida en una directiva del preprocesador #define.

#### System ( )

Es una función del lenguaje de programación C incluida en su biblioteca estándar, dentro de la cabecera <stdlib.h>. Sirve para ejecutar subprocessos o comandos del sistema operativo.

«System» permite ejecutar a su vez otras funciones como: «cls», «dir» o «pause»

#### OBJETIVO(S):

Diferenciar e implementar diferentes tipos de datos en el lenguaje de programación C++ para distintos fines de uso de los datos.

Usar variables y constantes en C++ para la manipulación de los datos.

Utilizar comandos de la librería para el control de consola, entrada y salida en el lenguaje de programación C++ .



## REQUISITOS:

Elaborar el programa analizado en clase en el IDE Dev C++ que contiene el compilador de C++ bajo las siguientes características:

1. Realizar un encabezado de datos institucionales, personales y de la unidad de aprendizaje en modo comentario /\* \*/
2. Posteriormente capturar el código en c++ en el orden mostrado en clase
3. Realizar un reporte de práctica en el formato oficial en word adjunto en esta asignación con el contenido siguiente:
  1. **PORTADA.** Con datos de identificación: Nombre de la Universidad y del Centro Universitario, Nombre del alumno, nombre de la UA, Número y nombre de la práctica, Licenciatura, semestre, Docente: Dra. en T.I.E. Adriana Bustamante Almaraz.
  2. En la primera sección de **INTRODUCCIÓN** como parte del marco teórico, investiga los conceptos teóricos de las instrucciones vistas en clase para reforzar la inducción teórica. Los conceptos a investigar están escritos en letras mayúsculas dentro del código del programa coloca la definición de cada uno de sus conceptos. Así mismo coloca también como fundamento teórico la instrucción System para asignar el color de fondo y texto a la salida del programa.
  3. Para la sección de **DESARROLLO** de la práctica modifica el color de fondo y texto a la salida del programa a los colores de tu elección y complementa el programa con otras tres conversiones de moneda mexicana a moneda extranjera diferentes a los que están en el código de ejemplo, las constantes a utilizar de estos nuevos tipos de cambio colócalos en la sección de variables y constantes abajo del main(). Obtén las capturas de pantalla del código previamente capturado en DEV C++, la ejecución del programa y el resultado deseado después de la ejecución.
  4. Al finalizar coloca una sección de **CONCLUSIÓN PERSONAL** que contenga una redacción de tu experiencia con el uso y manejo de DEV C++ describe lo que aprendiste, la experiencia que tuviste, ¿Que fue lo más fácil y qué lo mas difícil que comprendiste de estos temas?
  5. En esta última sección de **RECURSOS EN LÍNEA** coloca los links proporcionados para el desarrollo de esta práctica, así mismo si utilizaste otros recursos digitales colócalos en esta sección.

La práctica se entrega en formato PDF con todos los requerimientos arriba mencionados.



**NOTA IMPORTANTE:** No olvides hacer entrega el archivo \*.CPP que genera Dev C++ con el código desarrollado en el mismo. Finalmente recuerda que la práctica es individual en caso de plagio en cualquiera de sus secciones la práctica se invalida.



#### **MATERIAL Y EQUIPO A UTILIZAR:**

El material y equipo a utilizar para realizar la práctica es el siguiente:

- PC con IDE Dev C++ versión 5.11 instalado
- Manual de prácticas
- Bibliografía recomendada en este manual



#### **DESARROLLO:**

1. Ingresa la librería `stdlib.h` para utilizar la instrucción `system`, la cual implementaremos para el cambio de color de fondo y letra; al inicio del código de programa de la siguiente forma:

```
include<stdio.h>
```

2. Para generar el código de programación de conversión de monedas extranjeras utilizar la directiva `#define` como constantes globales con el valor de las monedas a convertir a lo largo de la ejecución del programa.



## Universidad Autónoma del Estado de México

```
D:\estructura de datos\MONEDAS_EXTRANJERAS.cpp - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
Proyecto Clases(Fun) MONEDAS_EXTRANJERAS.cpp
1  /*****
2  /*UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO */
3  /*CENTRO UNIVERSITARIO VALLE DE TEOHUACAN */
4  /*LICENCIATURA EN INFORMATICA ADMINISTRATIVA 2º SEM */
5  /*UNIDAD DE APRENDIZAJE: ESTRUCTURA DE DATOS */
6  /*NOMBRE DEL ALUMNO: ANGEL LARA GONZALEZ */
7  /*DOCENTE: Dra. en TIE. ADRIANA BUSTAMANTE ALVARAZ */
8  /*PRACTICA 2 */
9  /*FECHA DE ETREGA: MARZO 22 DEL 2022 */
10 /*****
11
12 //¿Que es una directiva de preprocesador?
13 //R=Todos las directivas del preprocesador empiezan con #, y solo pueden aparecer espacios en blanco antes de una directiva del preprocesador en una línea.
14 //Las directivas del preprocesador se procesan por completo antes de que empiece la compilación.
15 //¿Que es directiva #include?
16 //R=Esta directiva hace que se incluya una copia de un archivo especifico en lugar de la directiva.
17
18 #include<iostream> //librerias
19 #include<stdlib.h>
20 #include<conio.h>
21 #include<stdio.h>
22
23 //¿Que es la directiva #define?
24 //R=La directiva del preprocesador #define crea constantes simbólicas (constantes representadas como símbolos) y macros (operaciones definidas como símbolos).
25 //Nota: esta sección se incluye para beneficio de los programadores de C++ que necesiten trabajar con código heredado de C. En C++,
26 //Las macros se pueden reemplazar comúnmente por las plantillas y funciones en línea.
27 //Una macro es una operación definida en una directiva del preprocesador #define.
28
29 #define Dolar_USA 20.53 //constantes globales
30 #define Euro 22.83
31 #define Dolar_Can 16.24
32
33 using namespace std;
34 using std::cout;//el programa usa cout
35 using std::cin;//el programa usa cin
36 using std::endl;//el programa usa endl
37
Compilador Recursos Registro de Compilación Depuración Resultados
Línea: 77 Col: 21 Sel: 0 Líneas: 77 Longitud: 3952 Insertar Done parsing in 0.016 seconds
05:48:41 PM 18/03/2022
```

- Después de declarar el inicio principal del programa mediante la función main declarar las variables y otras constantes en la sección de variables a utilizar para complementar las operaciones de conversión.



```
D:\estructura de datos\MONEDAS_EXTRANJERAS.cpp - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
Proyecto Clases(Fun) MONEDAS_EXTRANJERAS.cpp
38 int main()//inicio principal del programa
39 {
40     float peso, Cambio_DolarUSA, Cambio_Euro, Cambio_DolCan, Cambio_Yen, Cambio_PesoCol, Cambio_Yuan;//declaracion de variables
41     char usuario [40];//inserta una cadena de caracteres
42     float Libra_Esterlina=27.03;//otra manera de declarar constantes globales
43     float Yen=5.85;
44     float Peso_Col=187.54;
45     float Yuan=3.20;
46
47     system("cls");//limpia la pantalla antes de escribir nuevos mensajes
48     system("color 4A");//Asigna un color a la pantalla
49
50
51     cout<<"***convertidor de moneda Mexicana a Extranjera***\n";
52     cout<<"Introduce tu nombre de usuario:"; //printf)
53     cin.getline(usuario,40, '\n'); //se guardan todos los caracteres tecleados por el usuario (scanf)
54     cout<<"Inserte el valor en pesos mexicanos que desea convertir \n"; //solo los primeros 40 caracteres, hasta detectar el salto de linea
55     cout<<"$";
56     cin>>peso;
57     Cambio_DolarUSA=peso/Dolar_USA; //Es la representacion algoritmica de esta operacion
58     //Cambio_DolarUSA <- peso/Dolar_USA
59     Cambio_Euro=peso/Euro;
60     Cambio_DolCan=peso/Dolar_Can;
61     Cambio_Yen=peso/Yen;
62     Cambio_PesoCol=peso/Peso_Col;
63     Cambio_Yuan=peso/Yuan;
64
65     cout<<"La conversion de moneda Mexicana a dolar Americano es: " <<Cambio_DolarUSA<< endl;
66     cout<<"La conversion de moneda Mexicana a Euros es: " <<Cambio_Euro<< endl;
67     cout<<"La conversion de moneda Mexicana a dolar Canadiense es: " <<Cambio_DolCan<< endl;
68     cout<<"La conversion de moneda Mexicana a Yen es: " <<Cambio_Yen<< endl;
69     cout<<"La conversion de moneda Mexicana a Peso Colombiano es: " <<Cambio_PesoCol<< endl;
70     cout<<"La conversion de moneda Mexicana a Yuan es: " <<Cambio_Yuan<< endl;
71     cout<<"Usuario que realizo la transaccion: " <<usuario<< endl;
72
73     //Apartir de estas instrucciones se marca el fin del programa
74     getch();//se realiza una pausa dentro del programa, perteneciendo a la libreria conio.h

```

4. Después de la declaración de variables, escribe la instrucción:

system (color "color de fondo" "color de letra");

Donde en "color de fondo" y "color de letra" son números hexadecimales (De 0 a F) que equivalen a un color. Dicha equivalencia se encuentra en la siguiente tabla:

- 0 = Negro
- 1= Azul
- 2= Verde
- 3= Aguamarina
- 4= Rojo
- 5= Púrpura
- 6= Amarillo
- 7= Blanco
- 8= Gris
- 9= Azul Claro
- A= Verde Claro
- B= Aguamarina Claro
- C= Rojo Claro

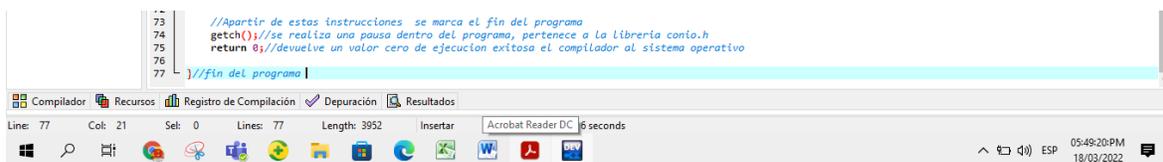
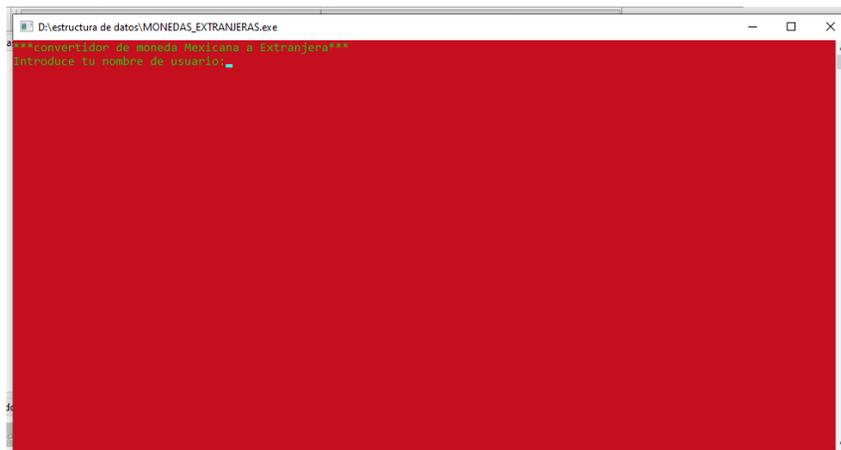


Universidad Autónoma del Estado de México

D= Púrpura Claro  
E= Amarillo Claro  
F= Blanco Brillante

Para generar un fondo de color azul y texto de color rojo, ejecutar un código de programa con la siguiente instrucción:

```
system("color 14");
```





Universidad Autónoma del Estado de México

(Ejecución del programa con características requeridas y los resultados deseados).

```
D:\estructura de datos\MONEDAS_EXTRANIERAS.exe
***Convertidor de moneda Mexicana a Extranjera***
Introduce tu nombre de usuario: ANGEL LARA GONZALEZ
Inserta el valor en pesos mexicanos que desea convertir:
2000
La conversión de moneda Mexicana a dolar Americano es: 97.9449
La conversión de moneda Mexicana a Euro es: 87.2394
La conversión de moneda Mexicana a dolar Canadiense es: 129.237
La conversión de moneda Mexicana a Yen es: 342.393
La conversión de moneda Mexicana a Peso Colombiano es: 10.6884
La conversión de moneda Mexicana a Yuan es: 625.010
Usuario que realizo la transacción: ANGEL LARA GONZALEZ

Process exited after 09.44 seconds with return value 0
Presione una tecla para continuar . . .
```

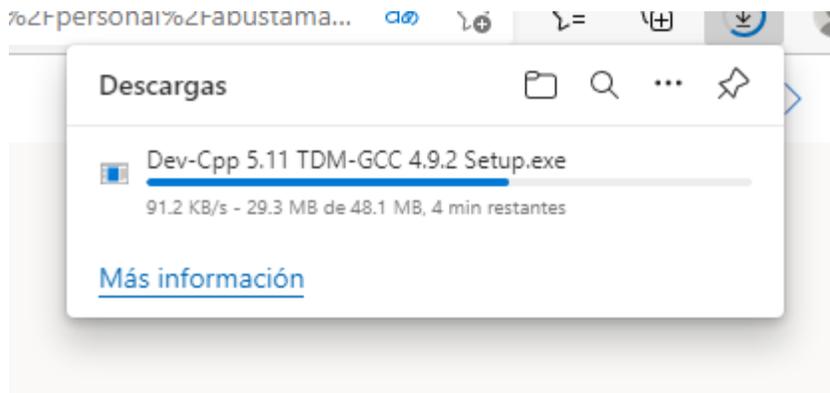
las

```
D:\estructura de datos\MONEDAS_EXTRANIERAS.exe
***Convertidor de moneda Mexicana a Extranjera***
Introduce tu nombre de usuario: ANGEL LARA GONZALEZ
Inserta el valor en pesos mexicanos que desea convertir:
2000
La conversión de moneda Mexicana a dolar Americano es: 97.9449
La conversión de moneda Mexicana a Euro es: 87.2394
La conversión de moneda Mexicana a dolar Canadiense es: 129.237
La conversión de moneda Mexicana a Yen es: 342.393
La conversión de moneda Mexicana a Peso Colombiano es: 10.6884
La conversión de moneda Mexicana a Yuan es: 625.010
Usuario que realizo la transacción: ANGEL LARA GONZALEZ
```



(Fin de la ejecución del programa, así como el resultado de salida de DEV C++).

### CONCLUSIONES:



En esta sección el alumno aprende a instalar el entorno de desarrollo integrado Dev C++ versión 5.11, comprende las secciones de un programa en lenguaje C++ a través de la escritura de un programa con salida de datos.

### CUESTIONARIO:

6. Con estas directivas del preprocesador se mandan a llamar las librerías de lenguaje C++ las cuales contienen las instrucciones clasificadas dentro de estas para escribir un código.
  - e) `#include <stdlib.h>`, `#include <conio.h>`
  - f) `#include <iostream>`, `#define`
  - g) `#define`, `const`, `void`
  - h) Ninguna de las anteriores



7. Con la siguiente línea de código en C++ se imprime un mensaje de salida al usuario en pantalla una vez compilado el programa correctamente:

- e) `cin>>usuario;`
- f) `cin.getline (usuario, 40,'\n');`
- g) `cout <<"dame el primer número:";`
- h) `cout<<"Introduce tu nombre de usuario:";`

8. Es la instrucción que permite al compilador devolver un valor al sistema operativo donde se ejecutó el programa para indicar que el proceso de compilación del código en C++ culminó correctamente:

- e) `Return n;`
- f) `return 0;`
- g) `VOID PROYECTO( );`
- h) `Int main ( )`

9. Es un programa informático que traduce un programa escrito en un lenguaje de programación, definido código fuente, a otro lenguaje de alto nivel:

- e) Código
- f) Lenguaje de programación
- g) Compilador
- h) Interprete

10. Conjunto de órdenes desarrolladas estructuradamente para que una computadora realice un conjunto de tareas determinadas:

- e) Lenguaje máquina
- f) Lenguaje de programación
- g) Programación
- h) Programa



## REFERENCIAS

Bustamante Almaraz, A. (09 de Febrero de 2021). *Microsoft stream canal personal de Adriana Bustamante Almaraz*. Obtenido de Mi primer programa en C++:  
<https://web.microsoftstream.com/video/26d407db-6617-413e-bbab->



Universidad Autónoma del Estado de México

ee524ee5a731?classId=05f2fefc-7678-4374-b4eb-  
6767b168d858&assignmentId=52a91ecd-a5e5-4a11-9646-bf2350cad999