



Universidad Autónoma de Querétaro

Facultad de Informática

C.U. a 13/02/2023

Estructuras de datos



Nombre del Alumno: Rafael Bárcena González

No. de Expediente: 307012

Correo electrónico: rbarcena14@alumnos.uaq.mx

Teléfono: 4424903125

Nombre del alumno: Diego Martínez García

Expediente: 307024

Correo electrónico: dmartinez188@alumnos.uaq.mx

Teléfono: 2712833731



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



En el mundo de la programación, el lenguaje C++ es uno de los más importantes y ampliamente utilizados en la industria y el mundo académico. En este documento, tendrás que desarrollar el índice de contenido propuesto:

El objetivo de este proyecto es proporcionarles la oportunidad de profundizar sus conocimientos sobre el lenguaje de programación C++ y desarrollar habilidades en la organización y presentación de información técnica. Al desarrollar este índice, tendrán una mejor comprensión de los conceptos clave de C++ y estarán preparados para seguir explorando y aprendiendo más sobre el lenguaje.

Introducción a C++

- ¿Qué es C++?

C++ es un lenguaje de programación que proviene de la extensión del lenguaje C para que pudiese manipular objetos. C++ significa “incremento de C”, dando a entender que se trata de una extensión del lenguaje de programación C. (*Qué es C++: Características y aplicaciones*, 2022)

- Historia y evolución de C++

A pesar de ser un lenguaje con muchos años, su gran potencia lo convierte en uno de los lenguajes de programación más demandados. Fue diseñado a mediados de los años 80 por el danés **Bjarne Stroustrup**. Su intención fue la de extender el lenguaje de programación C (*con mucho éxito en ese momento*) para que tuviese los mecanismos necesarios para manipular objetos. Por lo tanto, C++ contiene los paradigmas de la programación estructurada y orientada a objetos, por lo que se le conoce como un lenguaje de programación multiparadigma. (*Qué es C++: Características y aplicaciones*, 2022)

- Características y ventajas de C++

Las principales ventajas de programar en C++ son:

- **Alto rendimiento:** Es una de sus principales características, el alto rendimiento que ofrece. Esto es debido a que puede hacer llamadas directas al sistema operativo, es un lenguaje compilado para cada plataforma, posee gran variedad de parámetros de optimización y se integra de forma directa con el lenguaje ensamblador.
- **Lenguaje actualizado:** A pesar de que ya tiene muchos años, el lenguaje se ha ido actualizando, permitiendo crear, relacionar y operar con datos complejos y ha implementado múltiples patrones de diseño.
- **Multiplataforma**
- **Extendido:** C y C++ están muy extendidos. Casi cualquier programa o sistema están escritos o tienen alguna parte escrita en estos lenguajes (*desde un navegador web hasta el propio sistema operativo*). (*Qué es C++: Características y aplicaciones*, 2022)



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



Configuración del entorno de desarrollo

- Descarga e instalación de un compilador C++

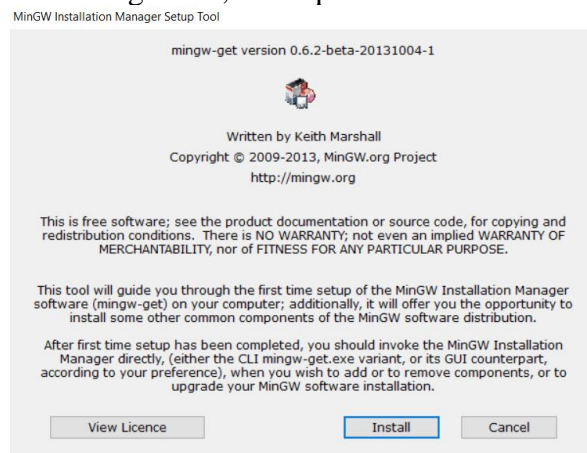
La entender la importancia de descargar un compilador en C++, hay que entender qué es un lenguaje compilado y qué es un lenguaje interpretado. Un lenguaje compilado es aquel lenguaje que para que pueda ser ejecutado, tiene que ser “traducido” al lenguaje máquina, es decir, el código fuente que escribe el programador (un código de C++, por ejemplo), es traducido a lenguaje máquina para su ejecución, en cambio, un lenguaje interpretado es aquel que no es traducido directamente a lenguaje máquina, sino que, un programa externo se encarga de leer el código, interpretarlo y ejecutarlo, sin necesidad de convertirlo a lenguaje máquina.

Como C++ es un lenguaje compilado, se necesita de un compilador que nos ayude a traducirlo a lenguaje máquina. De ahí la necesidad de instalar un compilador.

1. Para descargar un compilador, basta con ir al siguiente link y dar click en descargar (botón verde):

<https://sourceforge.net/projects/mingw/>

2. Posteriormente, al abrir el archivo que de descargó, saldrá una ventana como la siguiente, en la que se dará click al botón **install**.



3. A continuación, se mostrará una ventana en donde podrás elegir en qué ruta se desea que se instale el compilador.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1

Step 1: Specify Installation Preferences

Installation Directory
C:\MinGW Change

If you elect to change this, you are advised to avoid any choice of directory which includes white space within the absolute representation of its path name.

User Interface Options

Both command line and graphical options are available. The command line interface is always supported; the alternative only if you choose the following option to ...

☒ ... also install support for the graphical user interface.

Program shortcuts for launching the graphical user interface should be installed ...

☒ ... just for me (the current user), or ... ☐ ... for all users * ...

☒ ... in the start menu, and/or ... ☒ ... on the desktop.

* selection of this option requires administrative privilege.

View Licence Continue Cancel

4. Una vez terminado el proceso, la ventana lucirá algo así y se dará click en continuar.

MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1

Step 2: Download and Set Up MinGW Installation Manager

Download Progress

Catalogue update completed; please check 'Details' pane for errors.

Processed 113 of 113 items : 100 %

Details

mingw-get: *** INFO *** setup: unpacking mingw-get-setup-0.6.2-mingw32-beta-20131004-1-xm1.tar.xz
mingw-get: *** INFO *** setup: updating installation database
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-bin.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-gui1.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-11c.tar.xz
mingw-get: *** INFO *** setup: installation database updated

View Licence Continue Quit

5. Posteriormente se abrirá una nueva ventana en donde tendremos que dar click izquierdo a la opción que diga mingw32-gcc-g++ y dar click en **Mark for installation**

MinGW Installation Manager

Installation Package Settings

Package	Class	Installed Version	Repository Version	Description
mingw-developer-toolkit	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
mingw32-base	bin		2013072300	A Basic MinGW Installation
mingw32-gcc-ada	bin		6.3.0-1	The GNU Ada Compiler
mingw32-gcc-fortran	bin		6.3.0-1	The GNU FORTRAN Compiler
mingw32-gcc-g++	bin		6.3.0-1	The GNU C++ Compiler
mingw32-gcc-objc	bin		6.3.0-1	The GNU Objective-C Compiler
msys	bin		2013072300	A Basic MSYS Installation (meta)

Unmark
Mark for Installation
Mark for Reinstallation
Mark for Upgrade
Mark for Removal

General

The GNU C++ Compiler

This package provides the MinGW implementation of the GNU C++ language compiler. This is an optional component of the MinGW Compiler Suite; you require it only if you wish to compile programs written in the C++ language.

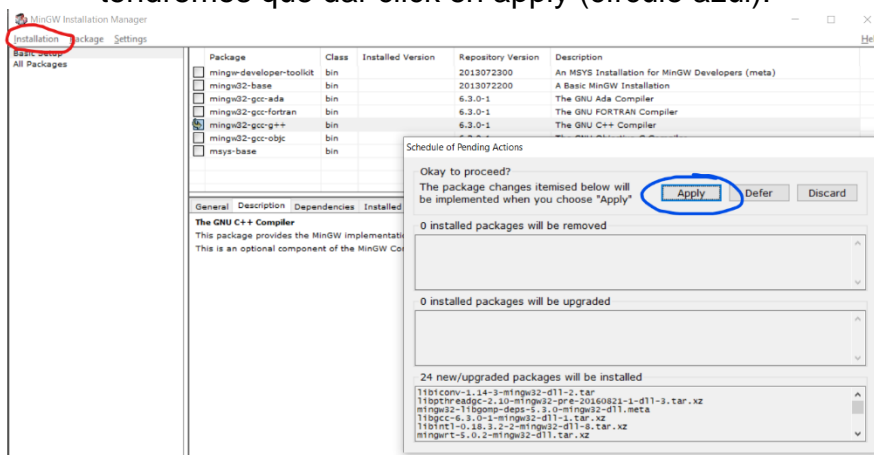


UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

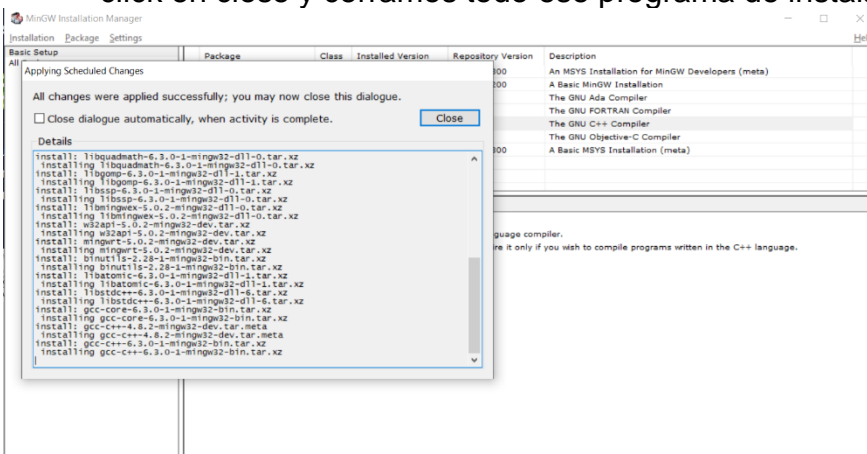
Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



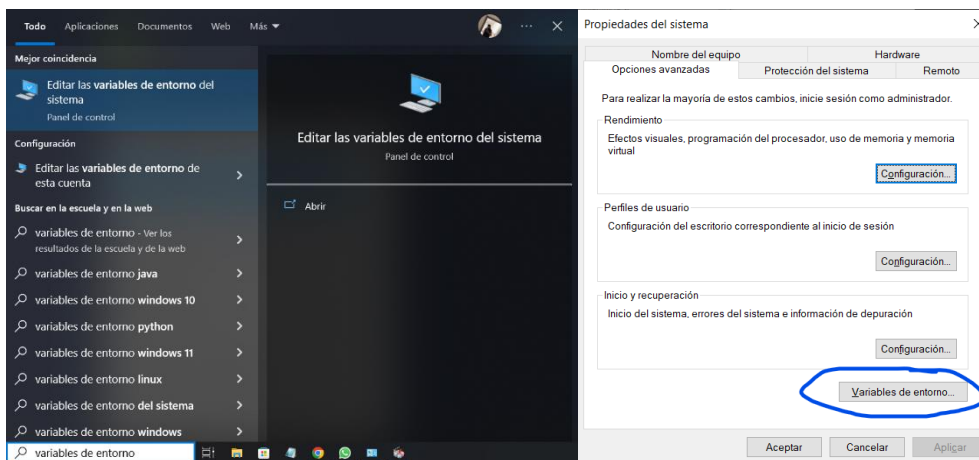
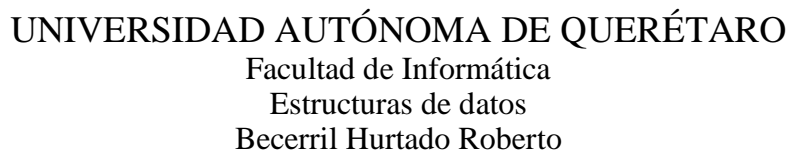
6. A continuación, se dará click en el botón installation (círculo rojo) y posteriormente, en la opción apply changes, se abrirá una ventana y solo tendremos que dar click en apply (círculo azul).



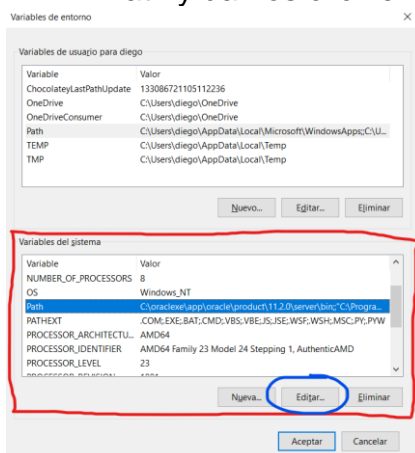
7. Cuando la descarga termine, saldrá una ventana como la siguiente, daremos click en close y cerramos todo ese programa de instalación.



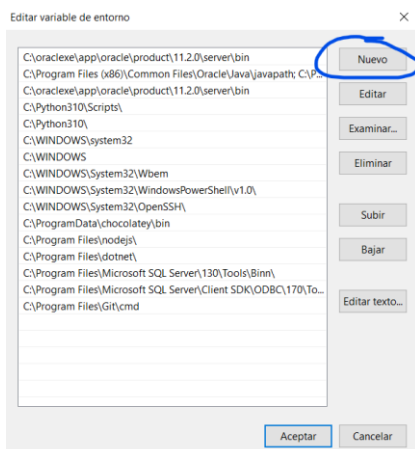
8. Posteriormente, buscamos en Windows “variables de entorno” y abrimos la opción variables de entorno (circulo azul):



9. Ya estando en la ventana de variables de entorno, vamos a la sección de variables del sistema (recuadro rojo), con un click seleccionamos la opción Path y damos click en editar (circulo azul).



10. Ahora en la nueva ventana, se dará click en nuevo (circulo azul).

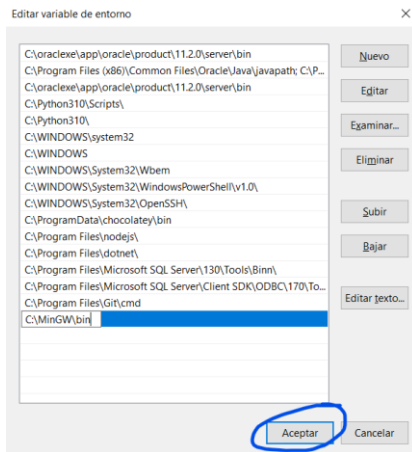


11. Ahora escribimos la ruta donde guardamos el programa "C:\MinGW\bin" y daremos click en el botón aceptar (circulo azul).



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



1. Para comprobar que todo esté bien, en CMD se escribe lo siguiente: `g++ --version`.

```
Simbolo del sistema
Microsoft Windows [Versión 10.0.19044.2486]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\diego>g++ --version
g++ (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Users\diego>
```

2.
 - Creación y ejecución de un programa básico en C++

`#include <iostream>`

`using namespace std;`

```
int main()
{
    cout << "Hola " << "mundo.";
}
```




UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



```
Estructura de datos > 1 > C++ 1.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Hola " << "mundo.";
8  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Semestre 4\Estructura de datos> cd "d:\Semestre 4\Estructura de datos\1\output"
PS D:\Semestre 4\Estructura de datos\1\output> & .\1.exe
Hola mundo.
PS D:\Semestre 4\Estructura de datos\1\output> []
```

Variables y tipos de datos en C++

- Declaración y asignación de variables

Primero se especifica el tipo de dato que almacenará la variable (int, float, char, algún tipo declarado por tí como una estructura, etc), el cual indicará cuáles son los valores que aceptará dicha variable. Posteriormente, se nombra a la variable seguida de un punto y coma, lo cual indica que la declaración ha terminado. (Pérez, 2021)

```
1 int x;
2 char firstLetter;
3 float positionX;
4 double price;
```

Para asignar valores a una variable en una gran variedad de lenguajes que incluye a C++ se usa el operador "=" seguido del valor que le daremos a la variable

```
char x = 'a'; // Declaramos y asignamos en la misma línea

int num; //Declaramos el entero en una línea
num = 5; //Le asignamos un valor en otra línea
```

- Tipos de datos básicos en C++ (int, float, double, char, etc.)
 - **bool**: El tipo de dato bool, tiene un tamaño de 8 bits y un rango entre 0 y 1, en pocas palabras es cero o es uno (falso o verdadero). Este tipo de dato es comúnmente usado en condicionales o variables que solo pueden tomar el valor de falso o verdadero.
 - **int**: El tipo de dato int, tiene un tamaño de 32 bits y un rango entre -2.147.483.648 y 2.147.483.647. Este tipo de dato es usado para números enteros (sin cifras decimales)
 - **short int**: Tiene un tamaño de 16 bits y un rango entre -32.768 y 32.767.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



- **float**: El tipo de dato float tiene un tamaño de 32 bits, es usado comúnmente en números con 6 o menos cifras decimales. Tiene un rango entre $1,17549 \times 10^{-38}$ hasta $3,40282 \times 10^{38}$.
- **double**: El tipo de dato double tiene un tamaño de 64 bits, es usado para números de menos de 15 cifras decimales. Tiene un rango entre $2,22507 \times 10^{-308}$ hasta $1,79769 \times 10^{308}$.
 - **long double**: Tiene un tamaño de 96 bits y una precisión de 18 cifras decimales. Tiene un rango entre $3,3621 \times 10^{-4932}$ hasta $1,18973 \times 10^{4932}$.
- **char**: Las variables de tipo char, en C y C++ son consideradas vectores y como quizá sabrás a los vectores se les debe declarar un tamaño máximo, entre corchetes "[]" lo cual restringe un poco al no saber qué tamaño podría llegar a tener una cadena de caracteres, y aunque hay formas de evadir esto, es bastante complicado. (González, 2021)
- **Uso de constantes y literales en C++**
Los valores asignados a cada variable constante se denominan literales. Generalmente, ambos términos, constantes y literales se usan indistintamente.

Los datos constantes, tienen un valor fijo a lo largo de la ejecución del programa, es decir, este valor no cambiará durante toda la ejecución de nuestro programa. No se puede cambiar. Las constantes son muy útiles para especificar el tamaño de un vector y algunas otras cosas como la facilidad de uso y la confiabilidad del código. (González, 2020)

Operadores en C++

- **Operadores aritméticos (+, -, *, /, %, etc.)**

Permiten realizar operaciones aritméticas.

+	Suma
-	Resta
*	Multiplicación
/	División
%	Residuo

- **Operadores de comparación (==, !=, >, <, >=, <=, etc.)**

==	Devuelve un true si sus operandos son iguales, un false en caso contrario.
!=	Devuelve un true si sus operandos son distintos, un false si son iguales.
>	Devuelve un true si el valor de la izquierda es mayor que el de la derecha.
<	Devuelve un true si el valor de la izquierda es menor que el de la derecha.
>=	Devuelve un true si el valor de la izquierda es mayor o igual que el de la derecha.
<=	Devuelve un true si el valor de la izquierda es menor o igual que el de la derecha.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



- Operadores lógicos (&&, ||, !, etc.)

!	Se encarga de negar el valor que se encuentre en su entrada: !false = true
&&	Se encarga de comparar dos valores, cuando ambos sean verdaderos, devuelve un true, en cualquier otro caso, devuelve false
	Se encarga de comparar dos valores, cuando ambos sean falsos, devuelve un false, en cualquier otro caso, devuelve true.

- Operadores de asignación (=, +=, -=, *=, /=, etc.)

Se utiliza para modificar el valor de la variable.

Operador	Equivalencia
=	Sirve para asignarle el valor de la derecha, a una variable de la izquierda.
+=	Sirve para aumentar el valor de la variable de la izquierda, por sí misma más otro valor.
-=	Sirve para disminuir el valor de la variable de la izquierda, por sí misma menos otro valor.
*=	Sirve para multiplicar el valor de la variable de la izquierda, por otro valor, es decir: $a = a * b$
/=	Sirve para dividir el valor de la variable de la izquierda, entre otro valor, es decir: $a = a / b$
%=	Sirve para obtener el residuo de la variable de la izquierda, entre otro valor, es decir: $a = a \% b$

Control de Flujo en C++

- Sentencia if

una condición if es una estructura que nos permite definir acciones que se realizarán cuando se cumplan ciertas condiciones, modificando así la ejecución de tareas en el programa según sea necesario. (González, 2020b)

La sintaxis de esta sentencia es bastante intuitiva

```
if(condición a evaluar) //Por ejemplo X <= 10
{
    ....
    ....
    Bloque de Instrucciones si se cumple la condición....
    ....
    ....
}
....
Bloque de Instrucciones restante DEL ALGORITMO....
....
```

- Sentencia switch

Una condición Switch es una estructura de control condicional que permite definir múltiples situaciones en las que se puede satisfacer cualquier variable, y qué acción



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



tomar en cualquiera de esas situaciones, o incluso determinar qué acción realizar si ninguna de ellas se cumple.

Una forma fácil de explicar el funcionamiento de esta sentencia es poniendo de ejemplo un programa que conste de un menú de tres opciones en la que cada opción represente un número, queremos entonces que de acuerdo a un número ingresado por el usuario ejecutemos una acción correspondiente y en caso de que no corresponda a ninguna de las posibles opciones, mostrar un mensaje de error cualquiera.

La sintaxis es la siguiente:

```
switch(opción) //donde opción es la variable a comparar
{
    case valor1: //Bloque de instrucciones 1;
        break;
    case valor2: //Bloque de instrucciones 2;
        break;
    case valor3: //Bloque de instrucciones 3;
        break;
    //Nótese que valor 1 2 y 3 son los valores que puede tomar la opción
    //la instrucción break es necesaria, para no ejecutar todos los casos.
    default: //Bloque de instrucciones por defecto;
        //default, es el bloque que se ejecuta en caso de que no se de ningún caso
}
```

- Sentencias for y while

- **For:** Los ciclos for son lo que se conoce como estructuras de control de flujo cíclicas o simplemente estructuras cíclicas, estos ciclos, como su nombre lo sugiere, nos permiten ejecutar una o varias líneas de código de forma iterativa, conociendo un valor específico inicial y otro valor final, además nos permiten determinar el tamaño del paso entre cada "giro" o iteración del ciclo.

En resumen, un ciclo for es una estructura de control iterativa, que nos permite ejecutar de manera repetitiva un bloque de instrucciones, conociendo previamente un valor de inicio, un tamaño de paso y un valor final para el ciclo. (González, 2019)

La sintaxis es bastante simple, solamente hay que tener bien claros los tres componentes del ciclo



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



```
for(int i = valor inicial; i <= valor final; i = i + paso)
{
    ....
    ....
    Bloque de Instrucciones....
    ....
    ....
}
```

- While: Los ciclos while son también una estructura cíclica, que nos permite ejecutar una o varias líneas de código de manera repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final que esperamos, los ciclos while, no dependen directamente de valores numéricos, sino de valores booleanos, es decir su ejecución depende del valor de verdad de una condición dada, verdadera o falso, nada más. (González, 2018)

La sintaxis es la siguiente:

```
while(condición de finalización) //por ejemplo numero == 100
{
    ....
    ....
    Bloque de Instrucciones....
    ....
    ....
}
```

- Uso de break y continue
 - Break: Esta instrucción finaliza la ejecución de un bucle o de la instrucción condicional y pasa a la siguiente instrucción (si existe alguna)
 - Continue: Fuerza la transferencia del control a la expresión de control del bucle contenedor do, for o while más pequeño.

Funciones en C++

- Declaración y llamado de funciones

Para definir una función en C++ se utiliza la siguiente estructura:

```
tipo nombre_funcion (tipo1 arg1, tipo2 arg2,...,tipo_n arg_n){
    sentencias
    return expresion;
}
```

- En el tipo de función, se pondrá el tipo de valor que se espera que la función regrese.
- En el nombre de la función, el nombre con el que la mandaremos a llamar posteriormente.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



- En los argumentos van aquellas variables con las que la función va a trabajar. Se pone el tipo de cada dato y el nombre del dato.
- Dentro de las llaves, se pone todo el proceso de trabajo de los argumentos adquiridos y un return (opcional).

Para llamar una función, se debe primero definirla (es decir, crear la función, sus argumentos, proceso, etc.) y posteriormente declararla, para declarar una función basta con poner la función antes de nuestra función principal o “main”. Hecho esto, se puede llamar a la función con su nombre y pasándole los argumentos que previamente se configuraron.

● Pasos de argumentos y valores de retorno

Para pasar argumentos a una función, se debe primero, configurar qué argumentos se espera que reciba una función, de qué tipo serán esos argumentos, y qué nombre llevarán para trabajar con ellos en el proceso de la función.

Hecho eso, al momento de llamar la función se debe pasar dichos argumentos dentro de los paréntesis en el mismo orden en que fueron creados los argumentos, es decir, si al hacer la función, el argumento 1 es un int, entonces a la hora de llamar la función, el primer dato que se pondrá en el paréntesis es un int, si el segundo argumento es un string, entonces también el segundo argumento a la hora de llamar la función será un string.

Para retornar un valor de una función, basta con usar la palabra reservada return seguido del contenido que deseamos regresar (el cual se especifica a la hora de definir la función) finalizando con un punto y coma.

● Uso de funciones predefinidas y propias

Para hacer uso de las funciones predefinidas y propias, primero se tendrá que incluir la librería requerida con un #include <nombre de la librería>, posteriormente podremos hacer uso de dicha librería.

Para hacer operaciones matemáticas podemos usar #include <cmath>, con lo cual podremos hacer lo siguiente:

- acos(x): Arco coseno
- asin(x): Arco seno
- atan(x): Arco tangente
- atan2(y,x): Arco tangente de y/x (por si x o y son 0)
- ceil(x): El valor entero superior a x y más cercano a él
- cos(x): Coseno
- cosh(x): Coseno hiperbólico
- exp(x): Exponencial de x (e elevado a x)
- fabs(x): Valor absoluto
- floor(x): El mayor valor entero que es menor que x
- fmod(x,y): Resto de la división x/y
- log(x): Logaritmo natural (o neperiano, en base "e")
- log10(x): Logaritmo en base 10



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



- $\text{pow}(x,y)$: x elevado a y
- $\sin(x)$: Seno
- $\sinh(x)$: Seno hiperbólico
- $\text{sqrt}(x)$: Raíz cuadrada
- $\tan(x)$: Tangente
- $\tanh(x)$: Tangente hiperbólica

Arreglos y Cadenas en C++

- Declaración e inicialización de arreglos

Para declarar un arreglo es casi igual que para declarar una variable, la única diferencia es que necesitaremos saber el tipo de los datos que irán al interior de este, es decir, serán número enteros, o número decimales o cadenas de texto, etc.

```
tipo_de_dato nombre_del_vector[tamano];
```

Cuando tenemos declarado el arreglo, es posible asignarle valores y para esto hay varias formas de inicializar el arreglo

```
string vector[5] = {"5", "hola", "2.7", "8,9", "adios"};
```

Aquí especificamos el tamaño después del nombre del arreglo y los valores los ponemos entre comillas ya que son strings

```
int vector2[] = {1,2,3,4,10,9,80,70,19};
```

Aquí no especificamos el tamaño del arreglo por lo que tomara el tamaño del número de elementos que haya en la llave

NOTA: El índice del primer valor no es 1 si no que es 0

- Uso de ciclos for para recorrer arreglos

Para obtener todos los datos en un vector, debe iterar sobre la matriz o el vector, para iterar sobre él, casi siempre se usa un ciclo for y, en algunos casos más específicos, un ciclo while, pero generalmente un ciclo for es excelente para esto porque conocemos el tamaño del arreglo. La lógica de este proceso es la siguiente: el bucle for comenzará en cero y aumentará hasta el tamaño del vector, por lo que la variable de control que solemos llamar "i" será la variable que varía entre cero y el tamaño del vector, de esta forma al poner la i al interior de los corchetes, estaremos accediendo al valor de cada casilla del vector y podremos realizar las operaciones necesarias sobre dichos valores



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



```
using namespace std;

int main()
{
    int edades[] = {1,2,9,8,16,32,9,50,36,20,1,87};
    int limite = (sizeof(edades)/sizeof(edades[0]));
    for (int i = 0; i < limite; i++)
    {
        cout<<edades[i]<<endl;
    }
}
```

- Trabajo con cadenas en C++ (uso de char y string)

En C no existe un tipo predefinido para manipular cadenas de caracteres (string). Sin embargo, el estándar de C define algunas funciones de biblioteca para tratamiento de cadenas.

Una cadena en C es un array de caracteres de una dimensión (vector de caracteres) que termina con el carácter especial '\0' (cero).

El formato para declarar una cadena es:

char nombre[n];

donde: $n \geq 1$ y representa a la longitud-1 real de la cadena.

Un ejemplo de declaración de cadena:

char cadena [5];

Debido a que en la representación interna de una cadena de caracteres es terminada por el símbolo '\0', para un texto de "n" caracteres, debemos reservar "n+1". El carácter '\0', aunque pertenece a la cadena, no aparece al utilizar funciones como printf.

En el caso especial de los arrays de caracteres, podemos utilizar varias formas de inicialización:

char cadena[] = "Hola";

char cadena[] = {'H','o','l','a',0};

char cadena[] = {'H','o','l','a','\0'};

sin especificar el tamaño de la cadena, o especificando el tamaño:

char cadena[5] = "Hola";

char cadena[5] = {'H','o','l','a',0};

char cadena[5] = {'H','o','l','a','\0'};

Estructuras de Datos en C++

- Uso de estructuras (struct)

La palabra reservada struct indica que los elementos que vienen agrupados a continuación entre llaves componen una estructura.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



La diferencia entre una clase y una struct en C++ es que una clase crea sus miembros automáticamente de manera privada, mientras que en las struct, se crean public.

Para definir una struct en C++ se usa la palabra reservada struct, seguido del nombre de la estructura, posteriormente dentro de las llaves se ponen los tipos y nombre de las variables que contendrá la estructura.

- **Uso de punteros y referencias**

Una referencia (por ejemplo, un puntero) almacena la dirección de un objeto que se encuentra en otra parte de la memoria. A diferencia de un puntero, una referencia inicializada no puede hacer referencia a otro objeto ni establecerse en null. Hay dos tipos de referencias: referencias lvalue, que hacen referencia a una variable con nombre y referencias rvalue, que hacen referencia a un objeto temporal. El operador & significa una referencia lvalue, mientras que el operador && significa una referencia rvalue o una referencia universal (lvalue o rvalue) según el contexto.

- **Uso de clases y objetos en C++**

Para declarar una clase en C++ la estructura será la siguiente:

```
class Mi_Clase {  
    //atributos y funciones de la clase  
}; //punto y coma.
```

Para poder hacer la instancia o el objeto de una clase se utiliza la siguiente sintaxis:

```
Mi_Clase PrimeraClase;
```

Cabe mencionar que si se hace la instancia en el main, el objeto será local, y si se hace fuera del main, será un objeto global

Excepciones y Errores en C++

- **Uso de try y catch para manejar excepciones**

Para implementar el control de excepciones en C++, se usan las expresiones try, throw y catch.

En primer lugar, se debe usar un bloque try para incluir una o más instrucciones que pueden iniciar una excepción

Para controlar las excepciones que se puedan producir, implemente uno o varios bloques catch inmediatamente después de un bloque try. Cada bloque catch especifica el tipo de excepción que puede controlar. (Microsoft, 2023)

- **Uso de throw para lanzar excepciones**

Una expresión throw indica que se ha producido una condición excepcional, a menudo un error, en un bloque try. Se puede usar un objeto de cualquier tipo como operando de una expresión throw. Normalmente, este objeto se emplea para comunicar información sobre el error. (Microsoft, 2023)

- **Debugging y solución de errores en C++**

Debug o debugging es el nombre del proceso utilizado en la programación para identificar y eliminar cualquier error en el código del programa. Estos errores pueden deberse a algunos problemas de sintaxis o lógica.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



Muchos de los paquetes de lenguaje de programación incluyen su depurador, para que sea posible evaluar el código y obtener sugerencias para resolver los problemas

Para iniciar el depurador en C++ tenemos que presionar F5, Al pulsar F5, la aplicación se inicia con el depurador activado para analizar los procesos.

Al iniciar el proceso el programa se ejecuta paso a paso, para verificar los elementos que puedan representar un problema, para así lograr reconocer en que parte del código existe un error. Además, es posible utilizar el depurador para revisar un punto específico de código

```
11 int main()
12 {
13     std::vector<wchar_t> letters = { L'f', L'r', L'e', L'd', L' ', L's', L'm', L'i', L't', L'h' };
14     std::wstring name = L"";
15     std::vector<int> a(10);
16     std::wstring key = L"";
17
18     for (int i = 0; i < letters.size(); i++)
19     {
20         name += letters[i];
21         a[i] = i + 1;
22         SendMessage(name, a[i]);
23     }
24     std::wcin >> key;
25     return 0;
26 }
```

Los pasos que sigue la herramienta de debugging para encontrar el error en el código son:

- Analizar línea por línea el código
- Revisar la asignación de variables
- Interrumpir la asignación de un programa cuando se cumple un determinado tipo de parámetros

Conclusión y Recursos Adicionales

C++ es un lenguaje de programación de alto rendimiento y gran versatilidad que se utiliza en una amplia gama de aplicaciones y sectores, desde sistemas operativos y software de juegos hasta robótica y dispositivos de Internet de las cosas

Al realizar este índice de conocimientos básicos de C nos dimos cuenta de las similitudes y diferencias con otros lenguajes de programación, además, de aclarar dudas acerca de algunos temas.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



Así mismo, se conocieron nuevos conceptos que hacen referencia que se encuentran en muchos lenguajes de programación pero que en C++ funcionan de manera diferente, tal es el caso de las structs, las cuales hacen referencia a las clases en POO, pero que, en el caso de las structs, los atributos se inicializan de manera pública y no privada.

Igualmente, el presente trabajo nos ayuda a comprender mejor lo que es un lenguaje compilado y uno interpretado, dicha información fue necesaria explicarla ya que se requería instalar un compilador que nos ayude a correr los programas de C++.

Y por esto creemos que aprender C++ puede ser importante para los programadores que buscan desarrollar aplicaciones de alto rendimiento y eficiencia en una variedad de sectores, desde juegos y sistemas operativos hasta robótica y dispositivos

Referencias

Qué es C++: Características y aplicaciones. (2022, 2 noviembre). OpenWebinars.net.

Recuperado 15 de febrero de 2023, de <https://openwebinars.net/blog/que-es-cpp/>

Pérez, H. (2021, 27 febrero). ▷ ¿ **【Cómo se declara una variable】** en tu lenguaje de programación favorito? El Camino Dev. <https://elcamino.dev/como-se-declara-una-variable/>

González, J. D. M. (2021, 4 enero). *Tipos Primitivos.*

<https://www.programarya.com/Cursos/C++/Sistema-de-Tipos/Tipos-Primitivos>

González, J. D. M. (2020, 4 agosto). *Variables y Constantes.*

[https://www.programarya.com/Cursos/C++/Sistema-de-Tipos/Variables-y-](https://www.programarya.com/Cursos/C++/Sistema-de-Tipos/Variables-y-Constantes)

[Constantes](https://www.programarya.com/Cursos/C++/Sistema-de-Tipos/Variables-y-Constantes)

González, J. D. M. (2020b, agosto 21). *Condional if.*

<https://www.programarya.com/Cursos/C++/Condicionales/Condional-if>

González, J. D. M. (2021a, enero 3). *Condional switch.*

<https://www.programarya.com/Cursos/C++/Condicionales/Condional-switch>

González, J. D. M. (2019, 1 octubre). *Ciclo for.*

<https://www.programarya.com/Cursos/C++/Ciclos/Ciclo-for>



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



González, J. D. M. (2018, 20 marzo). *Ciclo while*.

<https://www.programarya.com/Cursos/C++/Ciclos/Ciclo-while>

C. (2023, 10 enero). *break (Instrucción) (C++)*. Microsoft Learn.

<https://learn.microsoft.com/es-es/cpp/cpp/break-statement-cpp?view=msvc-170>

C. (2023b, enero 10). *continue (Instrucción) (C++)*. Microsoft Learn.

<https://learn.microsoft.com/es-es/cpp/cpp/continue-statement-cpp?view=msvc-170>

González, J. D. M. (2021c, mayo 18). *Arrays y Vectores*.

<https://www.programarya.com/Cursos/C++/Estructuras-de-Datos/Arreglos-o-Vectores>

Microsoft. (2023, 15 febrero). *Instrucciones try, throw y catch (C++)*. Microsoft Learn.

<https://learn.microsoft.com/es-es/cpp/cpp/try-throw-and-catch-statements-cpp?view=msvc-170>

Debug: ¿Qué es y para qué sirve? (2022, 23 julio). Higo.io: Maximiza el flujo de efectivo de tu empresa. <https://higo.io/glosario-contable/d/debug-que-es-y-para-que-sirve/>

M. (2023c, enero 4). *Tutorial: Depuración de código de C++ - Visual Studio (Windows)*.

Microsoft Learn. <https://learn.microsoft.com/es-es/visualstudio/debugger/getting-started-with-the-debugger-cpp?view=vs-2022>

M. (2023c, enero 4). *Tutorial: Depuración de código de C++ - Visual Studio (Windows)*. Microsoft Learn.

<https://learn.microsoft.com/es-es/visualstudio/debugger/getting-started-with-the-debugger-cpp?view=vs-2022>

Funciones en C++ — Fundamentos de Programación en C++. (s. f.).

https://www2.eii.uva.es/fund_inf/cpp/temas/8_funciones/funciones.html



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
Facultad de Informática
Estructuras de datos
Becerril Hurtado Roberto



Operadores de asignación — Fundamentos de Programación en C++. (s. f.).

https://www2.eii.uva.es/fund_inf/cpp/temas/4_sentencias_simples/operadores_asignacion.html

Acosta, I. C. (2021, 18 enero). *OPERADORES LÓGICOS Y DE COMPARACIÓN EN C++*. Geek Electrónica. <https://geekelectronica.com/operadores-logicos-y-de-comparacion-en-c/>

C++ - 9.7. Algunas funciones predefinidas útiles - AprendeAProgramar.com. (s. f.). <https://www.aprendeaprogramar.com/cursos/verApartado.php?id=16901>

C. (2023, 10 enero). *struct (C++)*. Microsoft Learn. <https://learn.microsoft.com/es-es/cpp/cpp/struct-cpp?view=msvc-170>

Clases y Objetos en C++ - Clases y Objetos en C++ (Práctica 1). (s. f.). CodinGame. <https://www.codingame.com/playgrounds/50557/clases-y-objetos-en-c-practica-1/clases-y-objetos-en-c>

Platzi: Cursos online profesionales de tecnología. (s. f.). <https://platzi.com/clases/2372-c-plus-plus/39697-instalacion-en-windows/>