****

**Actividad [#2] – [Operaciones Básicas en C++]**

**[Lenguajes de Programación 1]**

**Ingeniería en Desarrollo de Software**

**Tutor: Miguel Ángel Rodríguez Vega**

**Alumno: Ricardo Rivas Rocha**

**Fecha: 4-Marzo-2023**

**Índice**

**Portada ……………………………………………. Página 1**

**Índice ………………………………. Página 2**

**Introducción ……………………………………. Página 3**

**Descripción …………………………………………. Página 4**

**Justificación …………………………………………. Página 5**

**Investigación ……………………………… Página 6 a 9**

**Desarrollo ………………………………………………. Página 10 a 11**

* **Código**
* **Prueba del Sistema**

**Conclusión ……………………………………. Página 12**

**Referencias …………………………………………… Página 13**

**Introducción**

**En este pequeño proyecto vamos a estar interactuando con operaciones matemáticas básicas como son la suma la resta la multiplicación y división en base a esto nos ayudara a mejorar mas lo que son aspectos matemáticos básicos en código y en que plataforma lo estaríamos codificando así es en c++.**

**Descripción**

**En esta descripción vamos a llevar a cabo un código con números enteros, variables y operaciones aritméticas esto nos permitirá trabajar de manera óptima con el código y empezar a jugar mas con este ya que tenemos que saber los principios básicos de esta rama que es la programación y así entender a las computadoras de igual manera.**

**Justificación**

**En este pequeño proyecto vamos a hacer un código que contenga operaciones básicas que ya sabemos que son sumas restas multiplicaciones y divisiones esto para que en el mundo de la programación se utilizan números para decir un ejemplo tienes un código que se te requiere actualizar un producto después de su venta o hacer una simple aplicación como calculadoras u otros programas orientados a los negocios.**

**Investigación**

**¿Cuáles son las operaciones básicas en C++?**

**int** valor = 1;

el código completo a ejecutar es

**#include <iostream>**

**using** **namespace** std;

**int** **main**() {

**int** valor = 1;

}

SUMAS

Es la operación más básica: sumar. Y dentro de la suma la más simple es sumar una variable a un literal. Un literal es un número específico, es decir, un número escrito directamente en el código:

**int** valor = 10;

**int** valor\_incrementado = valor + 1;

cout << "Valor final: " << valor\_incrementado;

Pero podemos sumar dos variables:

**int** unidad = 1;

**int** valor = 10;

**int** valor\_incrementado = valor + unidad;

cout << "Valor final: " << valor\_incrementado;

O algo bastante obvio, sumar dos literales:

**int** suma\_de\_goles = 2 + 1;

cout << "Goles en el partido: " << suma\_de\_goles;

RESTAS

La operación resta es tan sencilla como la de suma y por supuesto también acepta dos literales, dos variables o una variable y un literal (y viceversa):

**int** anio\_actual = 2020;

**int** anio\_nacimiento = 2007;

**int** edad = anio\_actual - anio\_nacimiento;

cout << "Tienes: " << edad << " anios";

MULTIPLICACION

La operación multiplicación no se expresa con una "x" sino con un asterisco: *\**.

**int** dinero = 10;

**int** dinero\_deseado = dinero \* 5;

cout << "Me han dado " << dinero << " euros para ir al cine pero me gustaria que me hubieran dado " << dinero\_deseado << " euros...";

DIVISION

La división es una operación complicada cuando se trata de variables de tipo int. Se expresa con la barra inclinada a la derecha: */*. Pero su dificultad es que el resultado es... ¡siempre un número entero! ¿Qué por qué? Pues porque una variable de tipo *int* sólo puede almacenar valores enteros:

**int** precio\_cine = 9;

**int** precio\_deseado = precio\_cine / 2; *// precio\_deseado almacenara el valor 4*

cout << "El cine cuesta " << precio\_cine << " euros pero me gustaria que costara " << precio\_deseado << " euros";

Así pues el resultado es la división euclidiana o entera. Por ejemplo: 5/2=2, 4/2=2, y en los negativos -5/2=-2.

RESIDUO

El residuo de una división se calcula directamente sin necesidad de hacer la división previamente, y en C++ se expresa con el carácter porcentaje: *%*. Veamos un ejemplo:

**int** cartas = 48;

**int** jugadores = 5;

**int** cartas\_sobrantes = cartas % jugadores;

cout << "Al repartir " << cartas << " cartas entre los " << jugadores << " jugadores que somos nos sobran " << cartas\_sobrantes << " cartas";

INCREMENTO / DECREMENTO

Una abreviación muy habitual sobre la suma es usar los operadores *++ y*--para incrementar en una unidad o decrementar en una unidad respectivamente. Este operador se puede usar inmediatamente antes o inmediatamente después de la variable:

**int** contador = 0;

contador++;

cout << contador << endl; *// Mostrara 1*

++contador;

cout << contador << endl; *// Mostrara 2*

contador--;

cout << contador << endl; *// Mostrara 1*

--contador;

cout << contador << endl; *// Mostrara 0*

**¿De que manera se pueden Ingresar datos en C++?**

Tal como mencioné hace un momento, la lectura de datos en C++ es bastante simple. Leer datos por teclado en C++ se hace usando el comando cin >> es importante notar el uso de los dos signos >> que son usados para controlar el flujo de datos. No te preocupes mucho por ellos, solo ten en cuenta que cada vez que vaya a usar la instrucción cin debes agregarle >> para quedar con un cin>>. Una manera muy sencilla de recordar esta instrucción es que in significa entrar y como estamos programando en C++ le añadimos la letra C al comienzo quedando así cin>> (sin olvidar los >>).

Veamos unos ejemplos simples para leer datos en C++. Recuerda como dije más arriba que lo ideal para leer datos es indicarle al usuario qué es lo que esperamos que ingrese por lo que en estos ejemplos usaremos también lo recién aprendido (mostrar texto por pantalla).

#include "iostream"

#include "string"

using namespace std;

int main()

{

cout << "Hola! Este es un ejemplo en C++" << "\n" << "Por favor ingrese su nombre:" << "\n";

//La instrucción \n es un salto de línea Mostrando los textos separados

string nombre;//En esta variable estará almacenado el nombre ingresado.

cin >> nombre; //Se lee el nombre

cout << "Bienvenido al sistema " << nombre << ". Gracias por usar nuestra aplicación" << "\n";

return 0;

}

**¿De que manera se pueden imprimir datos en C++?**

Mostrar texto por pantalla en C++ es muy simple. Para imprimir una salida de texto en C++ se hace uso de la instrucción cout, junto con <<. Es importante tener en cuenta que la instrucción cout siempre va acompañada de << para controlar el flujo de datos que sale. No te fijes mucho en ellos, solo ten siempre presente que cout viene acompañado de << para tener cout << como resultado.

#include "iostream"

using namespace std;

int main()

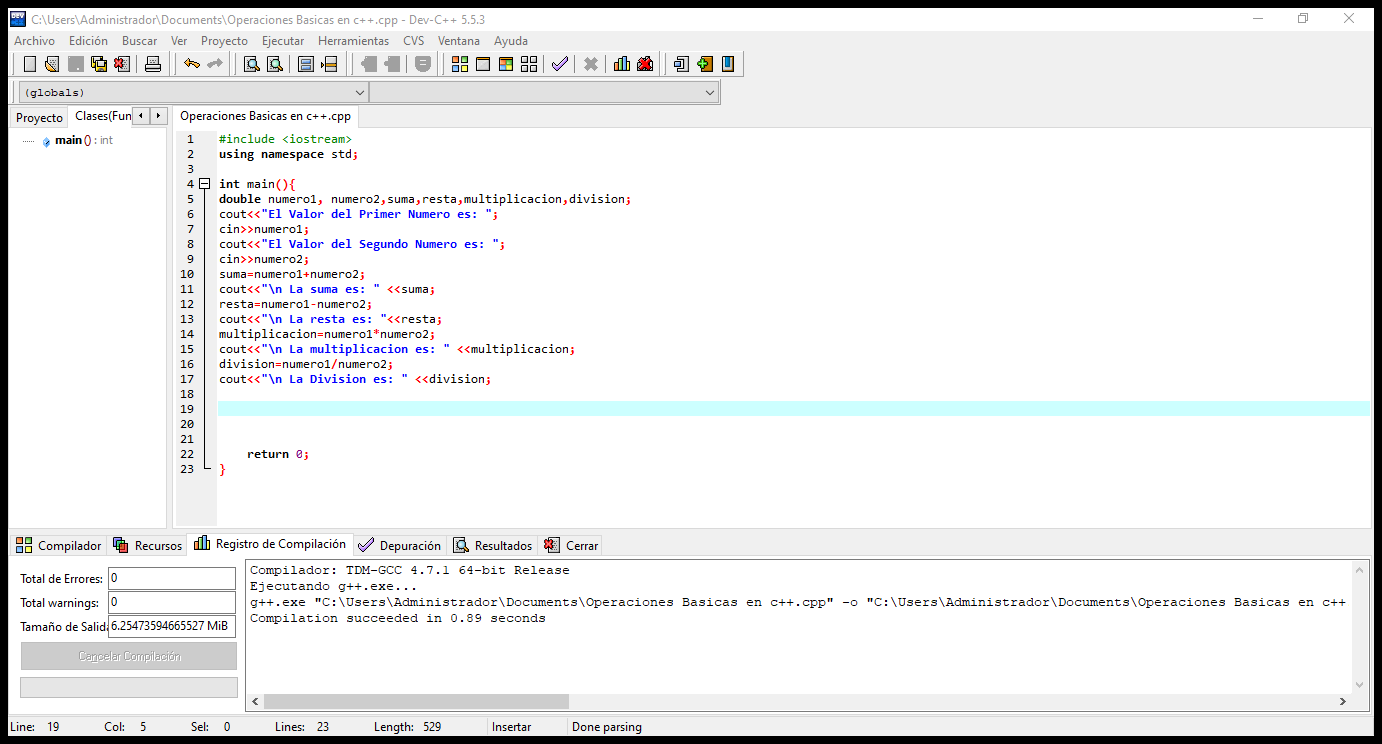
{

//Se muestra un mensaje por pantalla.

cout << "Hola Mundo" << " Desde ProgramarYa." << "\n";

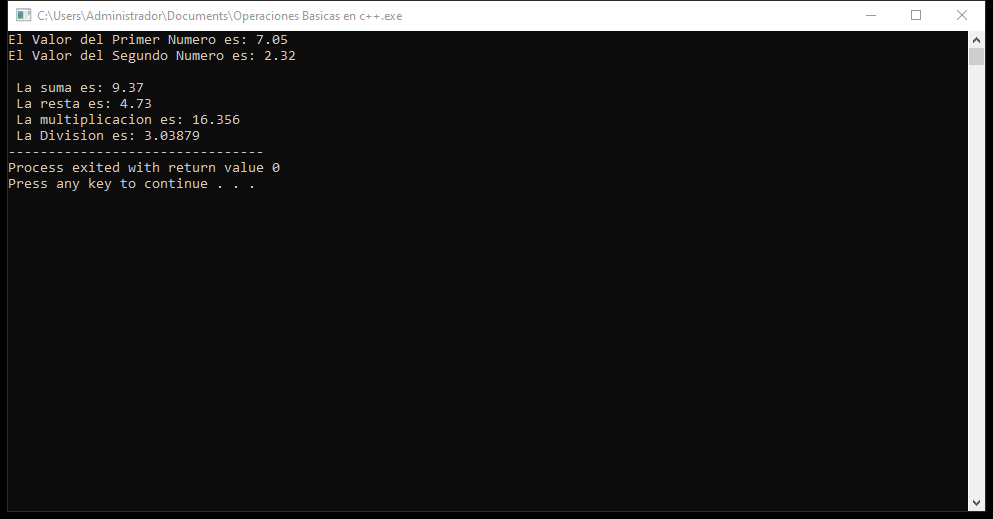
return 0;

}

**Desarrollo**

Como podemos observar aquí en esta parte está mi código lo que hice primero fue llamar a la librería luego de eso use el using namespace std; que nunca tiene que faltar y empezamos con int main que va a ser dentro de este donde vamos a ejecutar nuestro código declare double por que el ejercicio viene con decimales, aunque también se puede utilizar float

Declare mis variables que fueron mis operaciones básicas las imprimí con cout y las guarde con cin en caso de la individualidad del primer numero y el segundo así lo hice una vez que ya saque los valores de los 2 números en pantalla faltarían hacer las operaciones aritméticas básicas e imprimirlas así como a tener en cuenta los saltos de línea por que si tuve un poco de dificultad por que todo estaba corrido echando a perder se aprende jejeje y pues en mis operaciones imprima en pantalla con cout y guardaba con su respectiva variable el resultado en un momento saco una foto de pantalla en donde muestro mi código ya hecho.

**Prueba del Sistema**

Así es como sale mi captura de pantalla con el código en marcha vi algunos videos que también se puede imprimir en pantalla con printf y usar scanf para guardar los valores ya así yo veo más formas de programar y tengo otro enfoque en utilizar otro programa con estas mismas características.

**Conclusión**

**En esta breve conclusión hemos visto como es utilizar las condicionales if else y los operadores aritméticos en torno a c++ por lo que estoy muy contento y motivado de aprender este lenguaje por que al hacerlo tu te llena de emoción expresar tu código que funcione así para que también en un ambiente laboral sepamos como manejarlos y estudiar las librerías que estos lectores de código tengan se que la actividad 3 esta un poco mas orientada a un rfc se que habrá mas retos por pasar y pues bueno esto nos ha servido para reforzar mas estos lenguajes un saludo hasta la siguiente actividad.**

**Referencias**

(Jvilella, 2019)(programarya, 2023)

# Referencias

Jvilella. (23 de 09 de 2019). *aprende-olimpiada-informatica.org*. Obtenido de https://aprende.olimpiada-informatica.org/cpp-int-operaciones

programarya. (04 de 03 de 2023). *programarya*. Obtenido de https://www.programarya.com/Cursos/C++/Entrada-y-Salida-de-Datos