

Capítulo I Introducción a las Estructuras de Datos.

Contenido:

- Objetivo.
- Introducción.
- Estructuras de Datos.
- Programación Orientada a Objeto.
- Forma de Trabajo en la asignatura.
- Lenguajes de programación.
- Conclusiones.
- Tarea.
- Laboratorio.

Objetivo.

- Introducir a los estudiantes en las estructuras de datos y la forma de trabajo en la asignatura.

Introducción.

Cuando trabajamos en una aplicación cualquiera, por ejemplo, un procesador de texto como Word, y se va la luz, se pierde el documento que hasta ese momento estábamos escribiendo, si es que no lo hemos guardado en el disco duro. Esto ocurre debido a que estábamos escribiendo en la memoria interna de la computadora, también llamada temporal.

Para poder escribir en ella, fue necesario organizar la forma en que se iba a escribir, para poder acceder a esa información de forma rápida y oportuna. Objetivo que se logra con el uso de las estructuras de Datos. Aspecto a estudiar durante el transcurso de la asignatura.

Sin embargo, las aplicaciones que se desarrollan en el mundo real, por lo general trabajan con grandes volúmenes de información que deben ser almacenados de forma permanente y se requiere contar con un soporte que permita preservar esta información, la memoria periférica es una buena alternativa, también llamada memoria externa.

Estructura de Datos.

Que es una ED: Par (K,R).

- En la programación, se entiende una estructura de datos, como una forma de organizar un conjunto de datos elementales, con el objetivo de facilitar su manipulación.
- Por lo que se puede decir que una Estructura de Datos es:
 - Un par K,R.
 - Donde K representa a diversos elementos y R las relaciones que se establecen entre ellos.

Operaciones sobre las Estructuras de Datos.

- Sobre una estructura de Datos, se realizan diversas operaciones, estas contemplan:
 - Alta, adicionar ó insertar un nuevo elemento en la estructura.
 - Baja, borrar ó eliminar, un elemento de la estructura.
 - Búsqueda, encontrar un determinado elemento.
 - Otras operaciones, que se pueden realizar en la estructura de datos pueden ser:
 - Ordenamiento.
 - Apareo, donde dadas dos estructuras se obtienen otra nueva ordenada y que contenga a las apareadas.

Tipos de Estructuras de Datos:

- Lineales (cada nodo tiene un sucesor), por ejemplo, una Lista de Estudiantes.
- Ramificadas, por ejemplo, un Director puede tener asociado a varios subdirectores, y a su vez estos tiene asociados varios Jefes de departamento).
- Sobre cada una de estas ED, se pueden realizar diversas operaciones.

Estructuras de Datos a estudiar:

- Pilas.
- Colas.
- Listas Dinámicas.
- Multilistas.
- Árboles.
- Grafos.

Programación Orientado a Objetos:

Que es la POO.

El Mundo que nos rodea, es una colección de objetos que se relacionan entre sí.

El lenguaje natural está dotado de palabras, sustantivos, para nombrarlos y verbos para describir las operaciones o acciones que ellos desarrollan.

Es una técnica de estructuración, donde los objetos son los principales Elementos de Construcción que se conectan entre sí.

Es una Metodología de Programación de Propósito General que simula la forma en que el hombre trabaja.

Es una evolución natural de la programación. Un programa OO debe verse como un conjunto de Objetos en Comunicación.

Relación entre la POO y ED.

La clave de la programación OO es encontrar características comunes entre entidades implicadas en el problema y crear estructuras de datos (objetos) que capturen esas características comunes. Una vez identificados los objetos del problema, se definen sus características y se puede modelar su comportamiento.

Conceptos Básicos de la Programación Orientada a Objetos

- **Abstracción:**
 - Es aquella característica que nos permite identificar un objeto a través de sus aspectos conceptuales. Las características de los objetos pueden ser tan diferentes que puede

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

costarnos reconocer que pertenecen a una misma clase, sin embargo nosotros reconocemos a que clase pertenecen, gracias a la Abstracción.

- Ejemplos:
 - Un auto deportivo y otro familiar, son autos, ambos tiene rueda, volante, motor, puertas, etc.
 - Se tiene empleados administrativos y docentes, pero ambos gracias a la abstracción podemos decir que son Empleados, un objeto **Empleado** para gestionar el personal de la empresa, aunque se tengan diferentes tipos de empleados.
- **Clase.**
 - Es el conjunto de especificaciones o normas que definen como va a ser creado un objeto de un tipo determinado, algo parecido a un manual de instrucciones conteniendo indicaciones para crear el objeto.
 - Una clase constituye la representación abstracta de algo, mientras que un objeto constituye la representación concreta de lo que una clase define.
 - Determina el conjunto de puntos claves que ha de cumplir un objeto para ser considerado perteneciente a dicha clase o categoría, pues dos objetos de una misma clase no tiene que ser iguales, basta que cumplan las especificaciones de la clase.
- **Objeto**
 - Es la Unidad Básica de la Programación Orientada a Objetos.
 - Es una pieza que se ocupa de desempeñar un trabajo concreto dentro de una estructura organizativa de varios objetos, cada uno de los cuales ejerce la tarea particular para la que fue diseñada.
 - Es una agrupación de código, compuesta de propiedades y métodos que pueden ser manipulados como una unidad independiente.
 - Un objeto es un ejemplar de una clase.
- **Atributos:**
 - Definen los Datos del objeto permitiendo consultar o modificar su estado, en cualquier momento de acuerdo al procesamiento que se esté realizando.
 - Es lo primero que se declara dentro de la clase.

- **Métodos.**

- 1.- Son las Rutinas que definen el comportamiento de la clase

- 2.-Se tienen diversos tipos de métodos que a continuación se nombran.

- Constructores

- Solicitan y reservan espacio en memoria
 - Es el primer método que debe invocarse siempre.
 - Colocan un puntero a la zona de memoria donde están los métodos de la clase a la que corresponde el objeto, de manera que los mensajes que este reciba puedan ser atendidos adecuadamente.

- Procedimientos y Funciones

- Destruidores.

- Liberan la memoria que estaba siendo utilizada por un objeto que ya no se usará más.
 - Esto significa que la reservación de espacio en memoria para los objetos se realiza de manera dinámica, o sea, cuando se necesita espacio se reserva y cuando no se necesita más se libera.

- **Encapsulamiento.**

- Establece la separación entre el interfaz del objeto y su implementación.

- Por Ejemplo:

- Auto: Cuando se acelera, uno se limita a sólo eso, no se requiere saber la mecánica interna que hace que el auto se mueva más rápido o no.
 - En un sistema de gestión donde se necesite añadir un empleado, solamente se invoca al método añadir y él se encargará de hacer ese trabajo, no se necesita saber el código del método añadir.

- Ventajas del encapsulamiento:

- Permite ignorar detalles de almacenamiento: A envía un mensaje a B y para ello no necesita conocer los datos de B, ni como se almacenan estos.
 - Simplifica el programa. Cada objeto puede ser modificado sin afectar los módulos que lo acceden. Garantiza la integridad de los datos: el acceso a la

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

información almacenada en un objeto, o sea, la recuperación, inicialización, actualización o modificación de dicha información, debe realizarse siempre a través de los métodos definidos en la clase a la que pertenece dicho objeto.

- Proporciona seguridad al código de la clase.
- Simplifica la utilización de los objetos. Ya que un programador que use un objeto, no necesitará conocer los detalles de su implementación, se limitará a utilizarlo.

- **Instanciación.**

- Es el hecho de crear un objeto como tal, con características propias.

- **Polimorfismo**

- La palabra polimorfismo se deriva del griego poli: muchos y morfismo: forma, modalidad.
 - Determina que el mismo nombre de método, realizará diferentes acciones según el objeto sobre el que sea aplicado.
 - Ejemplo real:
Pelota y Vaso de Cristal: Si ejecutamos sobre ellos el método dejar caer, el resultado en ambos casos será diferente, mientras que el objeto pelota rebotará al llegar al suelo, el vaso cristal se romperá.
 - Ejemplo de programación:
Supongamos los objetos Ventana y Fichero, si ejecutamos sobre ambos el método abrir el resultado en ventana será la visualización de una ventana en el monitor, mientras que fichero se tomará un fichero en el equipo del usuario y se dejará listo para realizar sobre él operaciones de lectura o escritura.
 - Ventajas del Polimorfismo: Permite generar componentes reutilizables de alto nivel que pueden adaptarse para que se ajusten a diferentes aplicaciones mediante el cambio de sus partes de bajo nivel.

Relaciones entre Objetos.

- **Herencia**

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

- Mecanismo a través del cual se pueden construir clases en función de otras clases ya definidas.
- Establece que partiendo de una clase a la que denominamos clase base, padre o superclase, creamos una nueva clase denominada clase derivada, hija o subclase.
- En esta clase derivada disponemos de todo el código de la clase base, mas el nuevo código propio de la clase hija que escribamos para extender sus funcionalidades.
- A su vez podemos tomar una clase derivada y crear nuevas subclases a partir de ellas y así sucesivamente, componiendo lo que se denomina **jerarquía de clases**.
- En los lenguajes OO la herencia de atributos se extiende tanto a los datos o campos, como al comportamiento, o sea, los métodos.
- Se puede identificar a través del uso de la partícula **Es-un**.

- Por ejemplo, sean los objetos A y B si tiene sentido el enunciado “A es un B” se puede diseñar la clase A como una subclase de B.

- Ejemplo:

Supongamos la siguiente situación:

Se tiene Clase Base Empleado, pero surge la necesidad de realizar pago a empleados que no trabajan en la empresa, pues son comerciales que se pasan la mayor parte del tiempo desplazándose, y para realizar esto, utilizamos Internet, necesitando el número de tarjeta de crédito, la dirección de email, etc.

Esto se puede resolver creando la clase derivada TCiberEmpleado que heredará de la clase empleado y en la que solo se debe añadir las nuevas propiedades y métodos para las transacciones comerciales, ya que las demás operaciones se tienen en la clase TEmpleado de la cual ella hereda.

- Pertenencia.
 - Los objetos pueden estar formados a su vez por otros objetos.
 - Se dice en este caso que hay una relación de pertenencia puesto que existe un conjunto de objetos que pertenecen a otro objeto o se unen para formar otro objeto.
 - A este tipo de relación se le llama también contenedora.
 - Para reconocer esta relación entre objetos se puede realizar un análisis sintáctico sobre la misma usando la partícula (**tiene - un**).
 - Tiene-un también se puede expresar con Es-parte-de que se da también cuando alguno de los conceptos es un componente del otro, pero cuando los dos no son en sentido alguno, la misma cosa.

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

- **Utilización.**
- Hay situaciones en que un objeto utiliza a otro para realizar una determinada tarea, sin que ello suponga la existencia de una relación de pertenencia entre dichos objetos.
- Para saber si existe esta relación debemos realizar un análisis sintáctico sobre el mismo empleando la partícula “usa un”
- El código de una clase hace uso (utiliza) un objeto de otra clase.
- Por ejemplo
- “Un objeto ventana usa un objeto empleado”, devolvería verdadero.

Forma de Trabajo en la Asignatura:

- **Grupos y Horarios:**
 - Grupo1
Horario de Teoría: lunes de 14-16
 - Grupo2
Horario de Teoría: miércoles 16-18
- **Horarios de Laboratorios.**
 - Martes de 14-16 y 16-18
 - Miércoles de 14-16

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

- **Sistema de Evaluación.**

Plan	I
Evaluaciones	Puntajes
Parciales	30
Prácticas	10
Laboratorios	25
Final	35
Total	100 puntos

- **Distribución de Clases.**

- Clases Teóricas: Se expondrá cada tema, explicará los conceptos relacionados y pondrá algunos ejemplos relacionados.
- Clases de Laboratorio: Serán evaluativos, en cada uno de ellos se practicará lo aprendido de cada tema. Por lo general se dejarán ejercicios a realizar para el laboratorio, de forma que el estudiante pueda realizar con antelación.

Lenguajes de programación.

- Lenguaje C++: e utilizará este lenguaje para el trabajo en esta asignatura. Habrá un tema para su estudio, pero se irán dejando problemas a resolver y también en laboratorio, que permitan la adquisición de una buena habilidad para el trabajo con él.
- Compilador ZINJAL.
- Visual Studio: como entorno de desarrollo se utilizará visual studio .Net. y dentro de él. El lenguaje de programación c++.

Prácticas a realizar:

1. Utilizando C++ y visual Studio .NET implemente los siguientes códigos:

- A. Sumar dos números.
- B. Restar dos números.
- C. Multiplicar dos números.
- D. Dividir dos números.

Antes de comenzar, vamos a describir algunas de las características propias del lenguaje C++, que permitirán realizar estas operaciones, entre estas características se destacan:

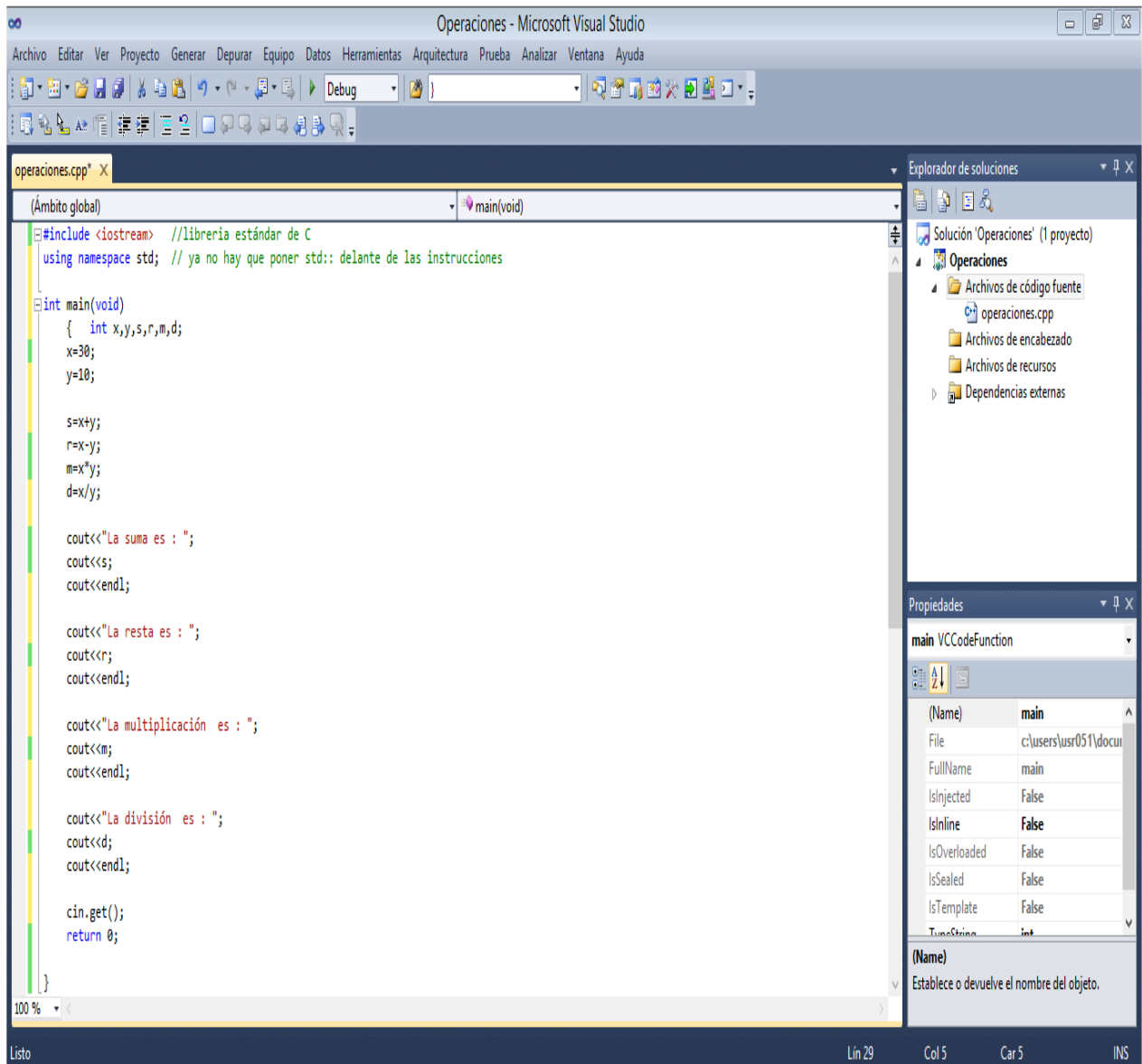
- Operadores Matemáticos ó aritméticos

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

- Entre estos operadores se tienen los de suma (+), resta (-), multiplicación (*) y división (/).
- Además resto ó módulo (%).
- Declaración de variables y tipos de Datos.
 - Se pone tipo de datos y luego el nombre de la variable
- Asignación de valores a las variables.
 - Se utiliza el operador =
- **Entrada de Datos por teclado y Salida en pantalla.**
 - Para la salida y entrada de datos estándar en c++ se tienen determinadas instrucciones y comandos.
 - Cuando se habla de entrada de Datos, significa que los datos se están entrando por teclado.
 - Cuando se habla de salida, se dice que los datos están escribiéndose en el monitor.
 - En c++ se tienen varias alternativas para ingresar y/o mostrar datos, esto depende de la librería que se esté utilizando; iostream.h ó sadio.h.
 - Veamos utilizando la librería iostream.h.
 - Esta biblioteca es una implementación orientada a objetos, basada en el concepto de flujos, que significa un medio de describir la secuencia de datos de una fuente a un destino o sumidero. Por ejemplo, cuando se introducen caracteres desde el teclado, se puede pensar en caracteres que fluyen o se trasladan desde el teclado a las estructuras de datos del programa.
 - Los objetos de flujo que vienen predefinidos entre otros son:
 - cin, para la entrada de datos por teclado.
 - `cin>>variable1>>...>>variable n;`
 - cout, salida de caracteres en la pantalla.
 - `cout<<variable1<<variable2<<...<<variable n;`

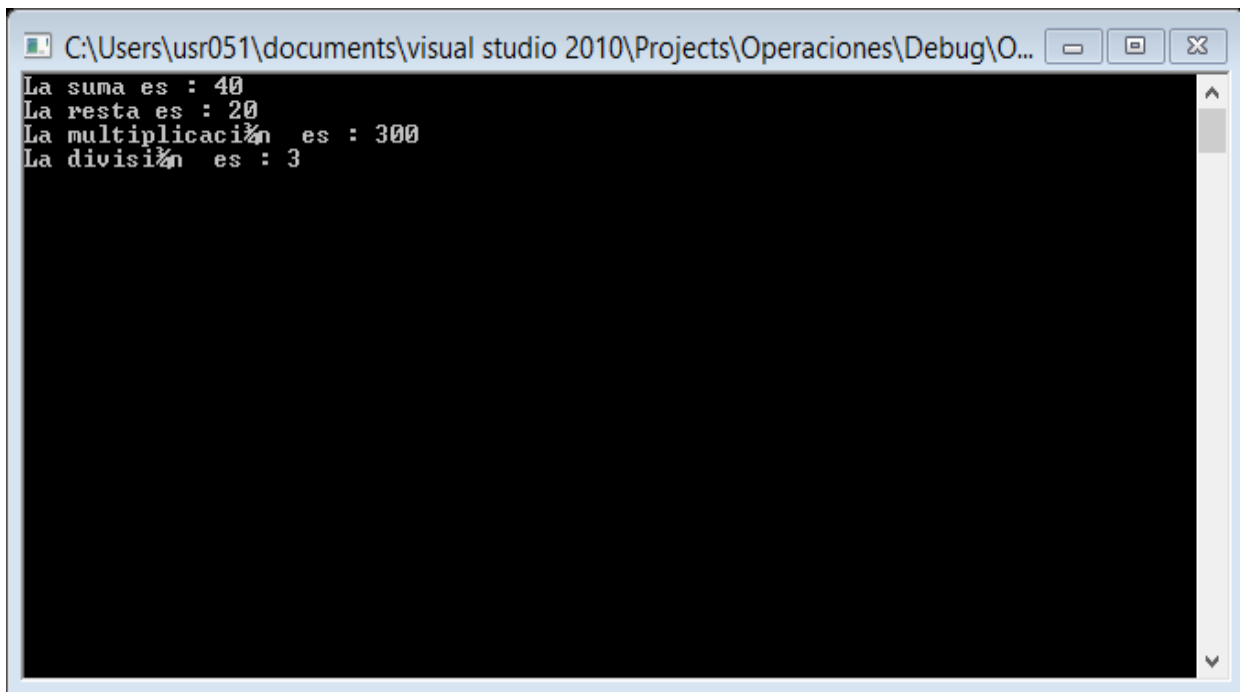
Desarrollo del Ejemplo:

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



El resultado de este programa será el siguiente:

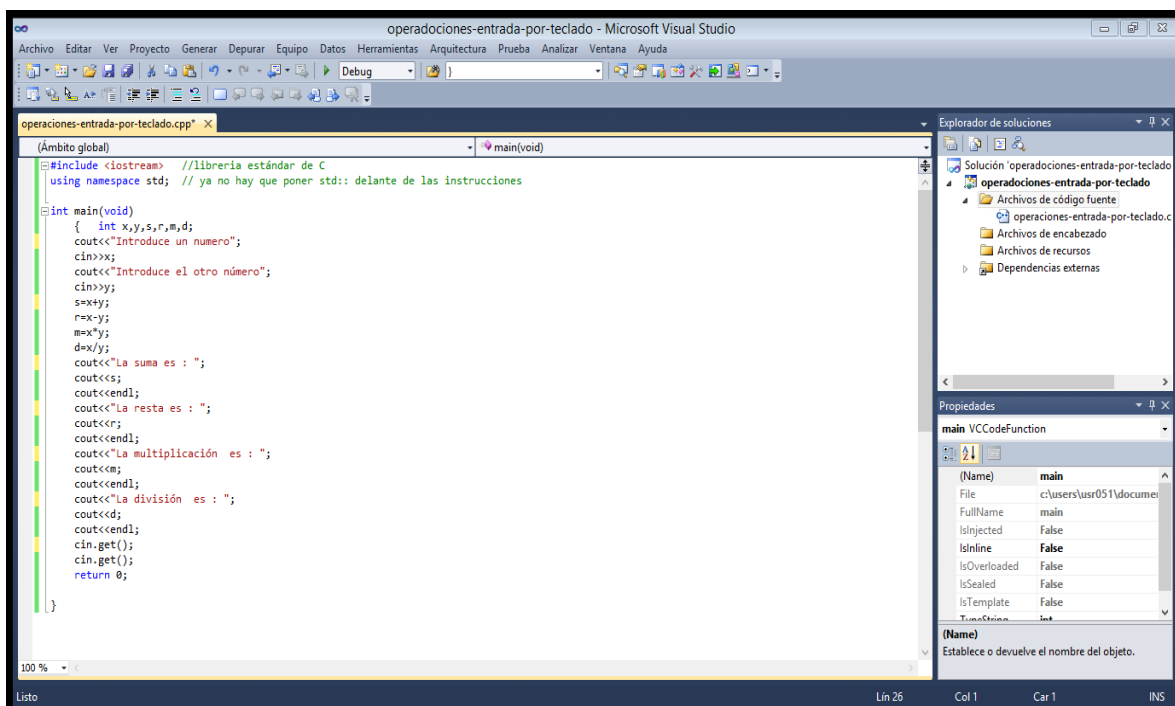
Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



```
C:\Users\usr051\documents\visual studio 2010\Projects\Operaciones\Debug\O...
La suma es : 40
La resta es : 20
La multiplicación es : 300
La división es : 3
```

Una vez realizada la actividad anterior, realizar lo mismo pero permitiendo la entrada de valores por teclado, es decir X y Y serán escrito por los usuarios y según estos valores así será el resultado:

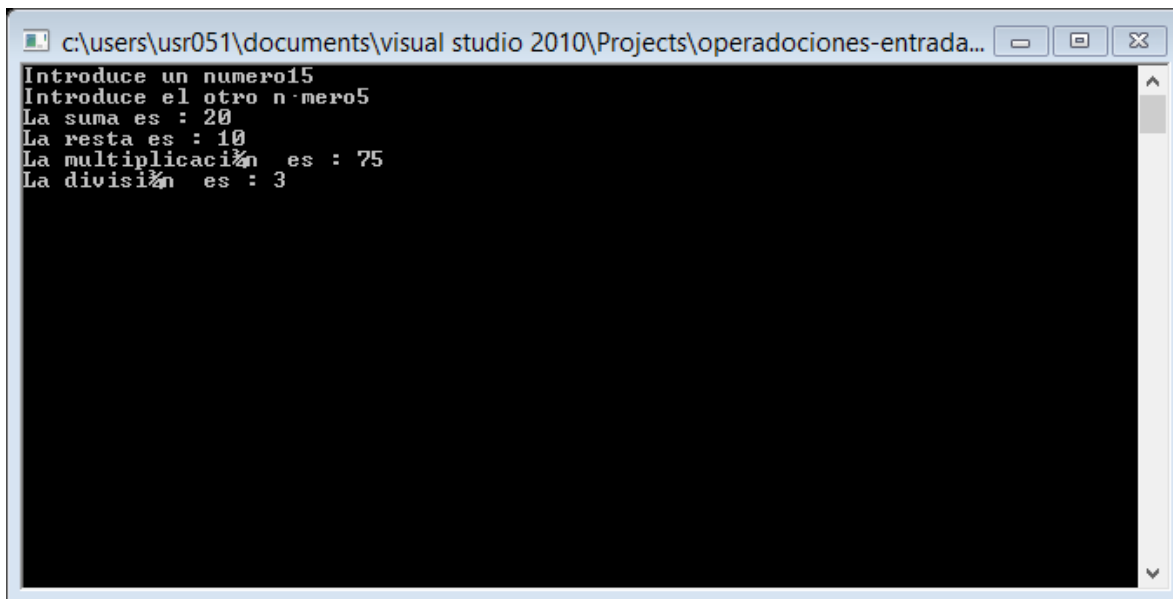
En este caso, el código quedará de la siguiente forma:



```
operaciones-entrada-por-teclado - Microsoft Visual Studio
Archivo  Editar  Ver  Proyecto  Generar  Depurar  Equipo  Datos  Herramientas  Arquitectura  Prueba  Analizar  Ventana  Ayuda
operaciones-entrada-por-teclado.cpp
(Ámbito global)
#include <iostream> //librería estándar de C
using namespace std; // ya no hay que poner std:: delante de las instrucciones

int main(void)
{
    int x,y,s,r,m,d;
    cout<<"Introduce un número";
    cin>>x;
    cout<<"Introduce el otro número";
    cin>>y;
    s=x+y;
    r=x-y;
    m=x*y;
    d=x/y;
    cout<<"La suma es : ";
    cout<<s;
    cout<<endl;
    cout<<"La resta es : ";
    cout<<r;
    cout<<endl;
    cout<<"La multiplicación es : ";
    cout<<m;
    cout<<endl;
    cout<<"La división es : ";
    cout<<d;
    cout<<endl;
    cin.get();
    cin.get();
    return 0;
}
```

El resultado se muestra a continuación:



```
c:\users\usr051\documents\visual studio 2010\Projects\operadociones-entrada...
Introduce un numero15
Introduce el otro numero5
La suma es : 20
La resta es : 10
La multiplicación es : 75
La división es : 3
```

2. Utilizando Visual Studio .NET y para comenzar a trabajar en la asignatura, es necesario trabajar con el lenguaje C++. Generar la primera aplicación que la llamaremos HOLA MUNDO, y qué a través de ella, explicaremos como se debe trabajar.

Para comenzar es necesario saber lo siguiente:

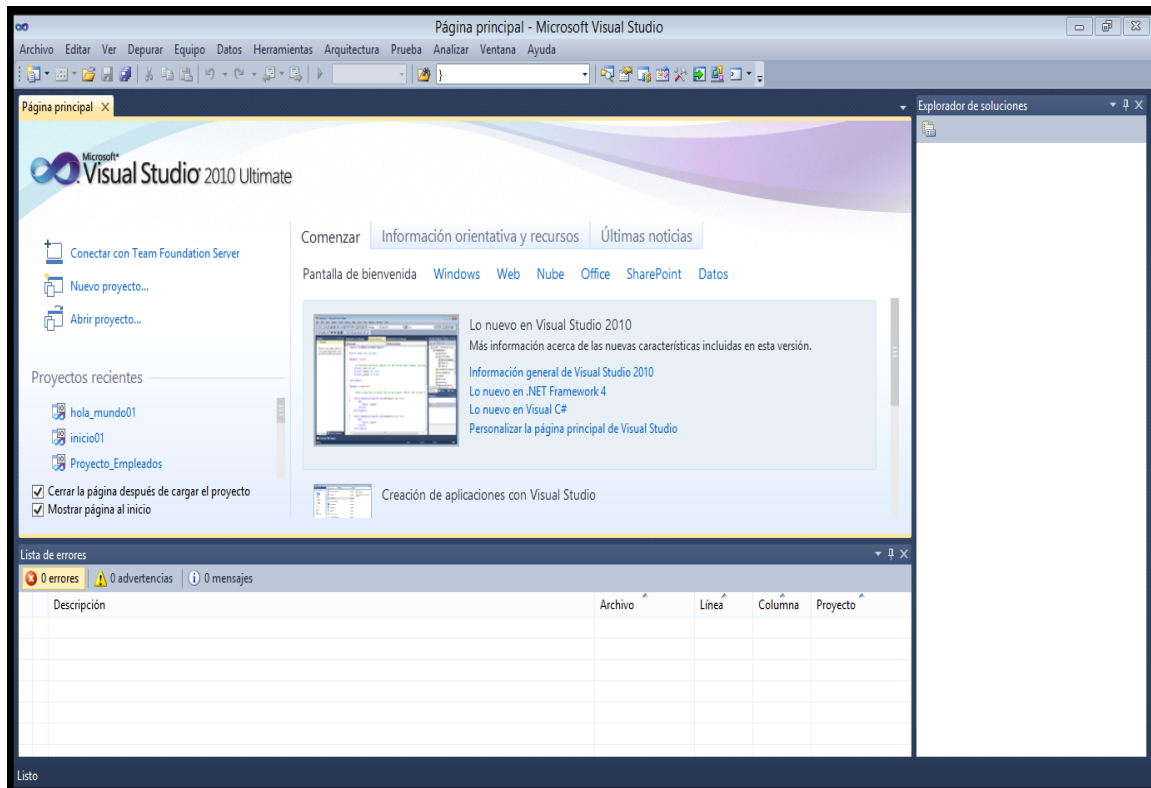
- 1.- Como abrir la aplicación Visual Studio .NET.
- 2.- Seleccionar el lenguaje de programación que vamos a utilizar, que en este caso será C++
- 3.- Escribir el programa principal y poner en uso las librerías necesarias para comenzar a trabajar.
- 4.- Escribir el Código del trabajo práctico a realizar.
- 5.- Correr la aplicación y ver el resultado de la misma.

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

A continuación, se describen cada uno de los puntos expuestos anteriormente:

1.- Como abrir la aplicación Visual Studio .NET.

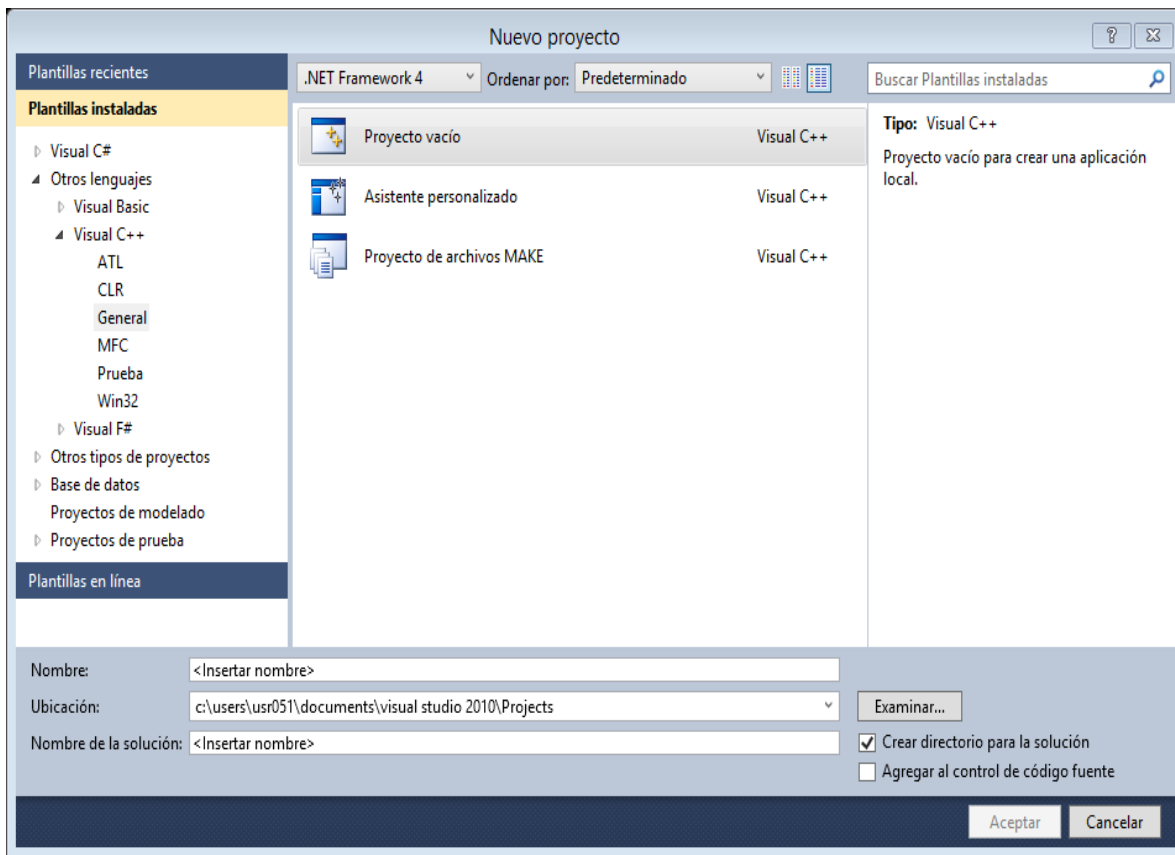
- Ir al menú inicio.
- Ir a la opción visual studio.NET.
- Aparecerá la siguiente pantalla:



2.- Seleccionar el lenguaje de programación que vamos a utilizar, que en este caso será C++.

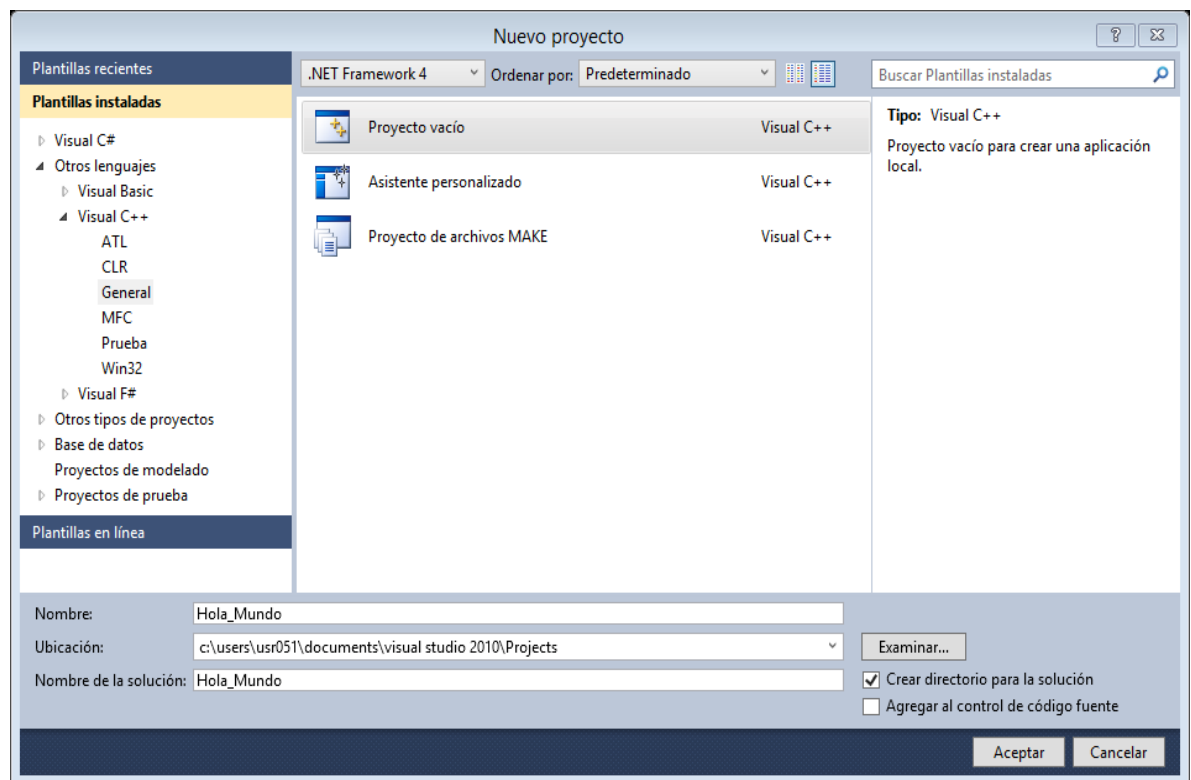
- Una Vez en la anterior pantalla de visual studio.NET, ir a la opción del menú Archivo, nuevo Proyecto, y aparecerá la siguiente ventana:

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



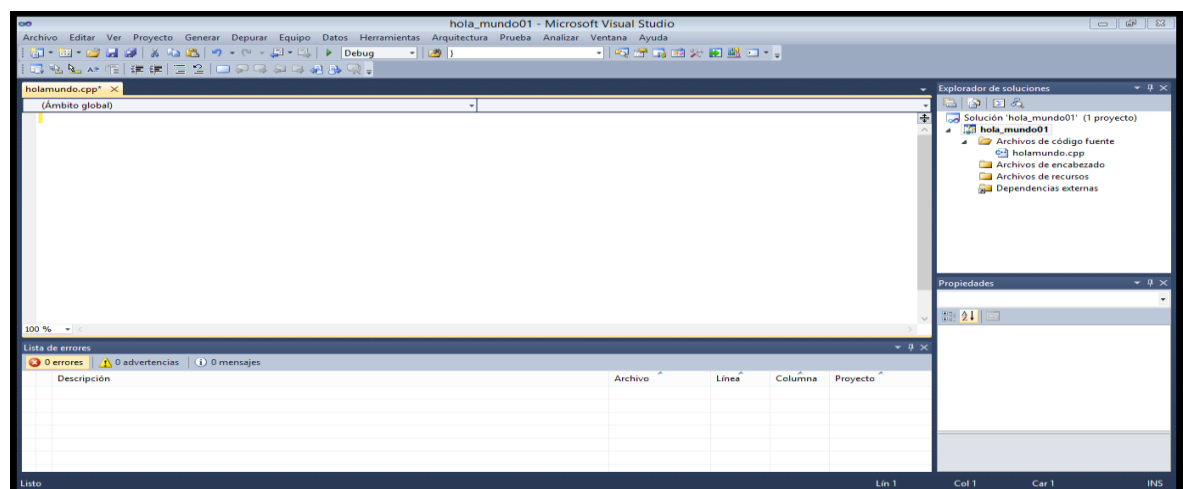
- Aquí seleccionaremos la opción general, proyecto vacío.
- Le pondremos nombre al proyecto
- Y haremos clic en el botón ACEPTAR.
- Si así lo desea, podrán puede cambiar la ubicación del proyecto.
- Quedando de esta manera:

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



3.- Escribir el Programa Principal y poner en uso las librerías necesarias para comenzar a trabajar.

- Una vez hecho aceptar, aparecerá una ventana.
- Deberá hacer clic derecho en el explorador de soluciones, archivo código fuente, agregar un nuevo elemento.
- Y aparecerá una ventana donde podrá escribir el código asociado al proyecto, como aparece a continuación:



Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

- En la ventana en blanco, deberá escribir el siguiente código, código siempre necesario para comenzar a trabajar:

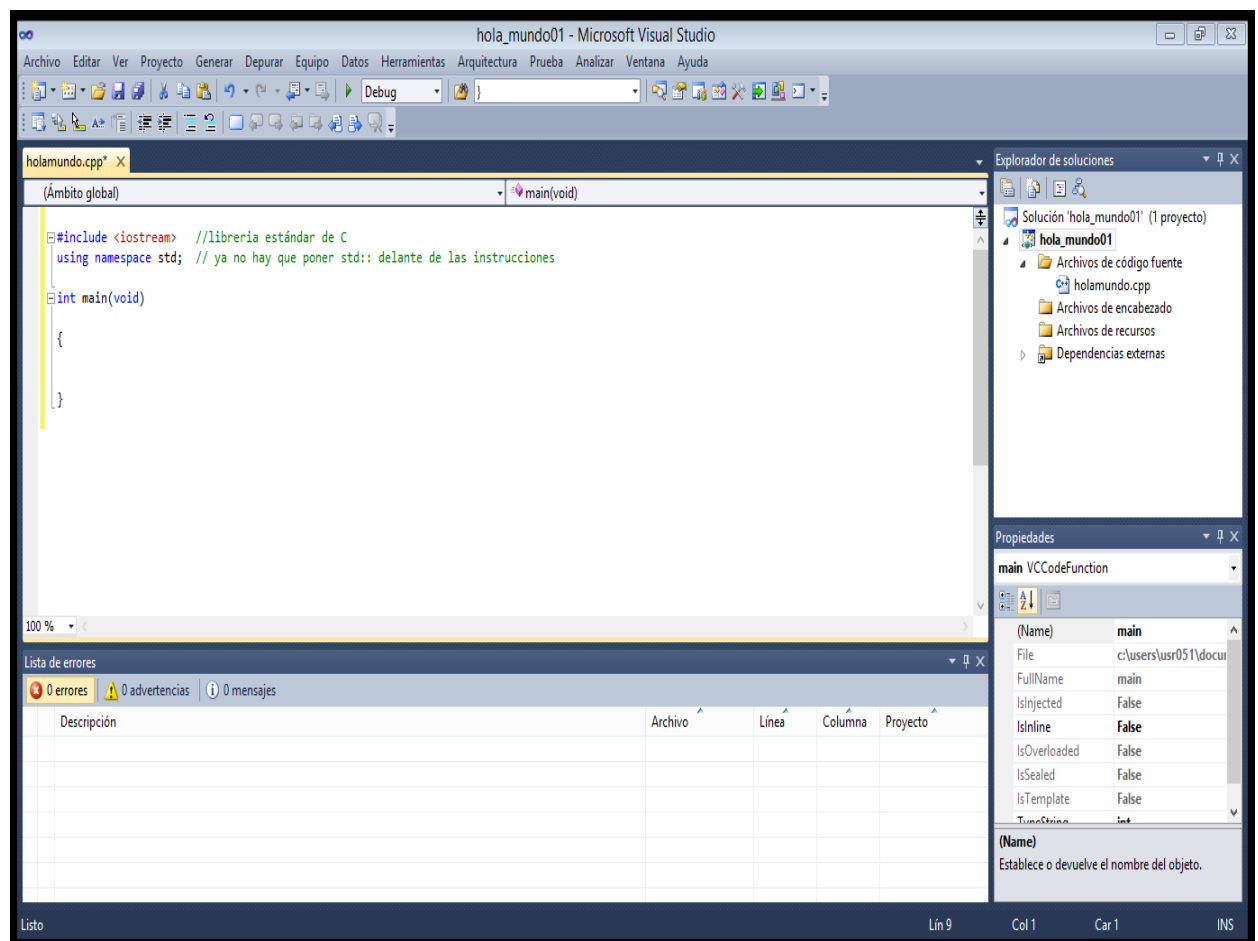
```
#include <iostream>    //librería estándar de C
using namespace std;  // ya no hay que poner std:: delante de las
instrucciones

int main(void)        // Este es el programa principal
{

}

}
```

- Quedando de la siguiente forma:



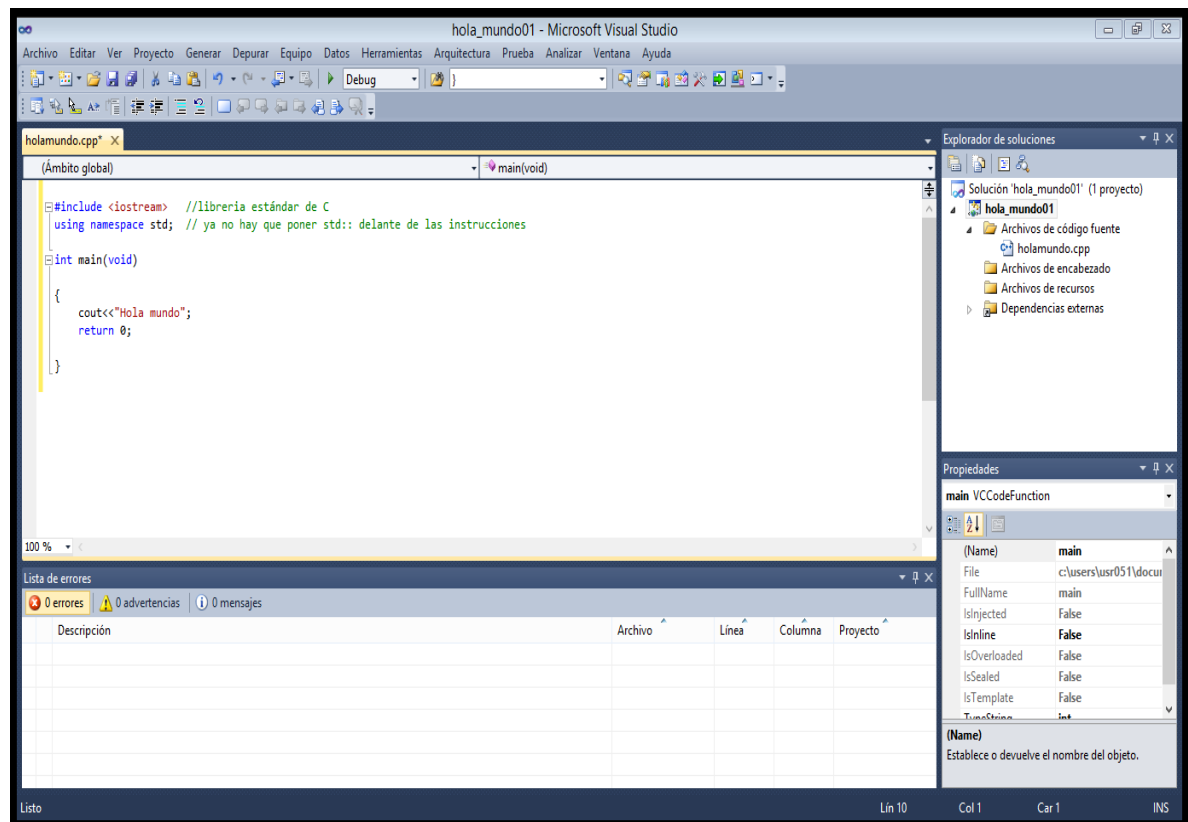
Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

4.- Escribir el Código del trabajo práctico a realizar.

- En el programa principal, entre las llaves, deberá escribir lo siguiente:

```
cout<<"Hola mundo";  
return 0;
```

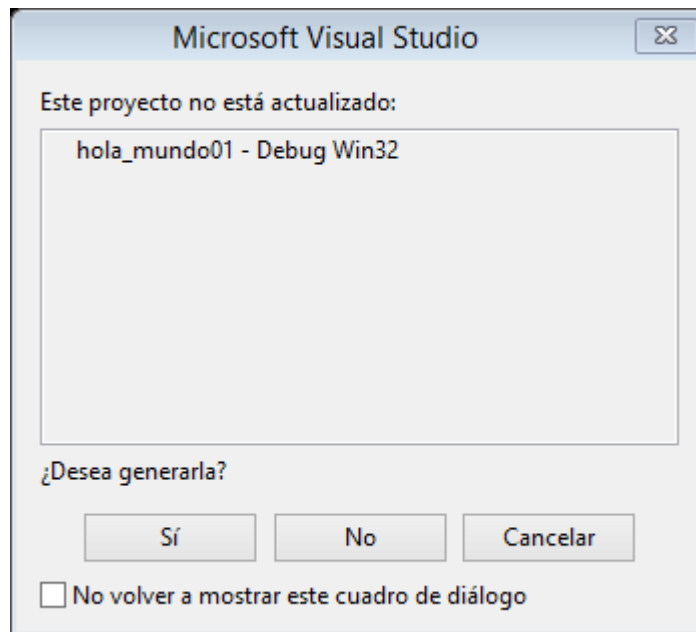
- Quedando de la siguiente forma:



5.- Correr la aplicación y ver el resultado de la misma.

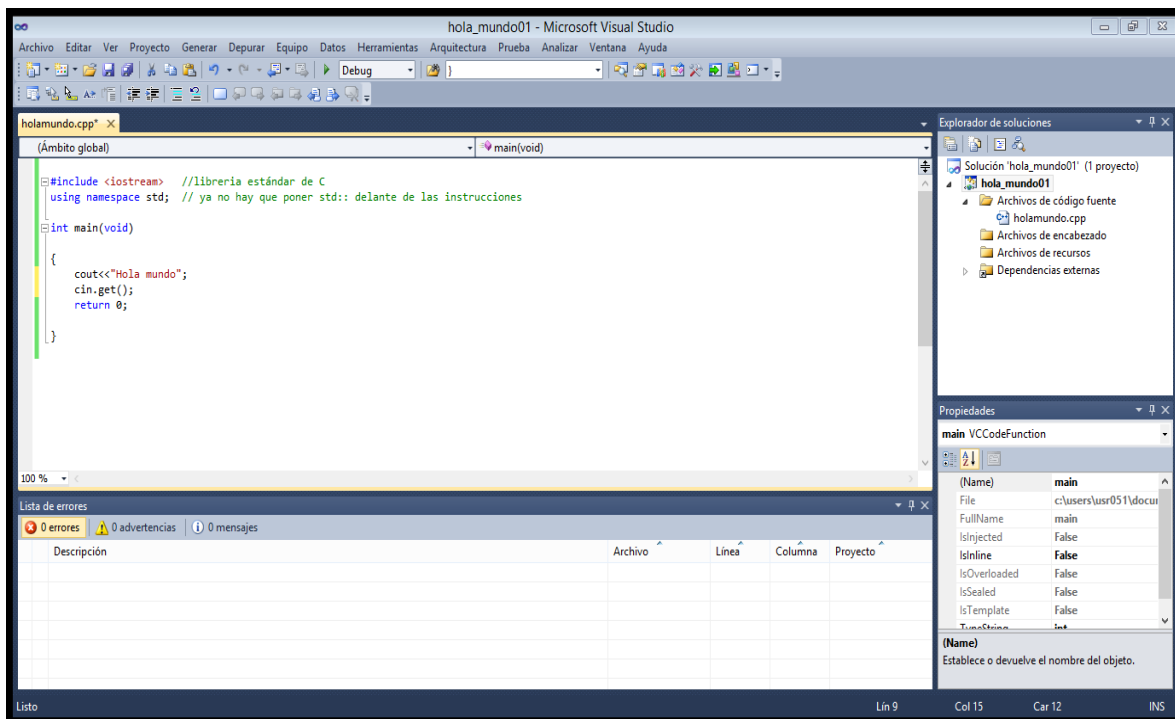
- En esta parte deberemos correr el programa, hacer play, iniciar depuración(F5).
- Aparecerá la siguiente ventana, donde pregunta ¿desea Generarla?, hará clic en SI.

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

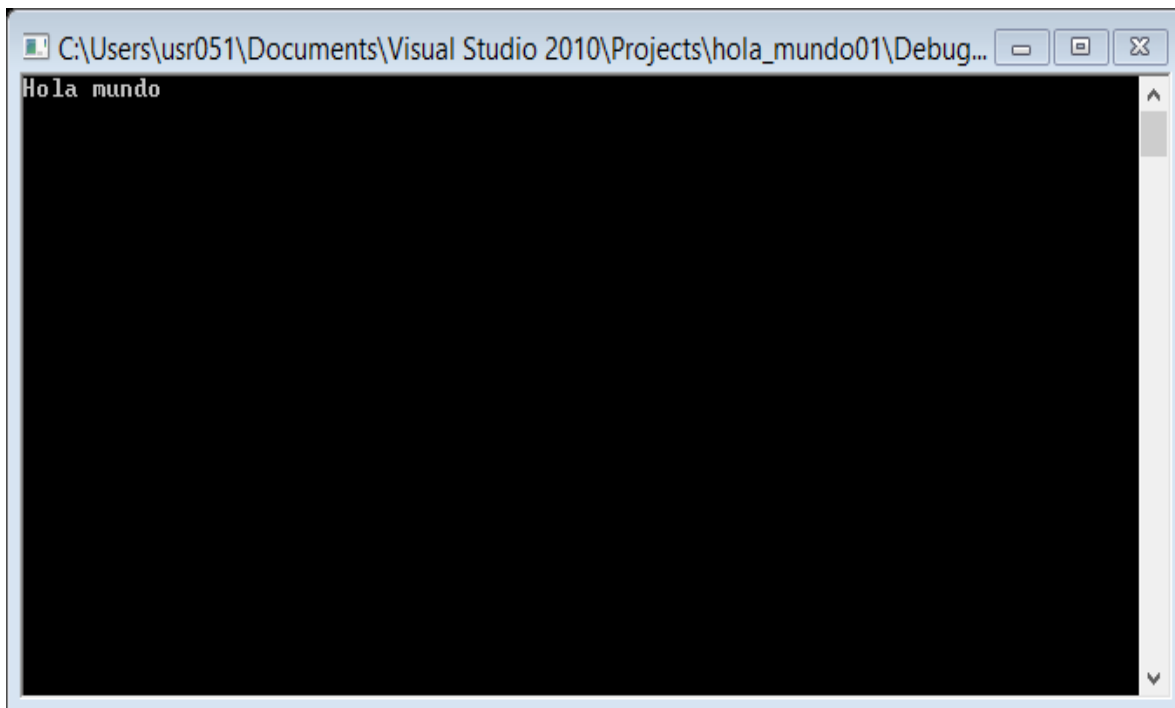


- Y se verá el resultado, este desaparece rápidamente por lo que podrá hacer lo siguiente que se explica.
- Adicionará este código al programa principal, para poder ver bien el resultado, esta instrucción se encarga de esperar por una tecla.
`cin.get();`
- Quedando de esta forma nuestro programa principal.

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



- Vuelva a correr y verá mejor el resultado, de esta forma:

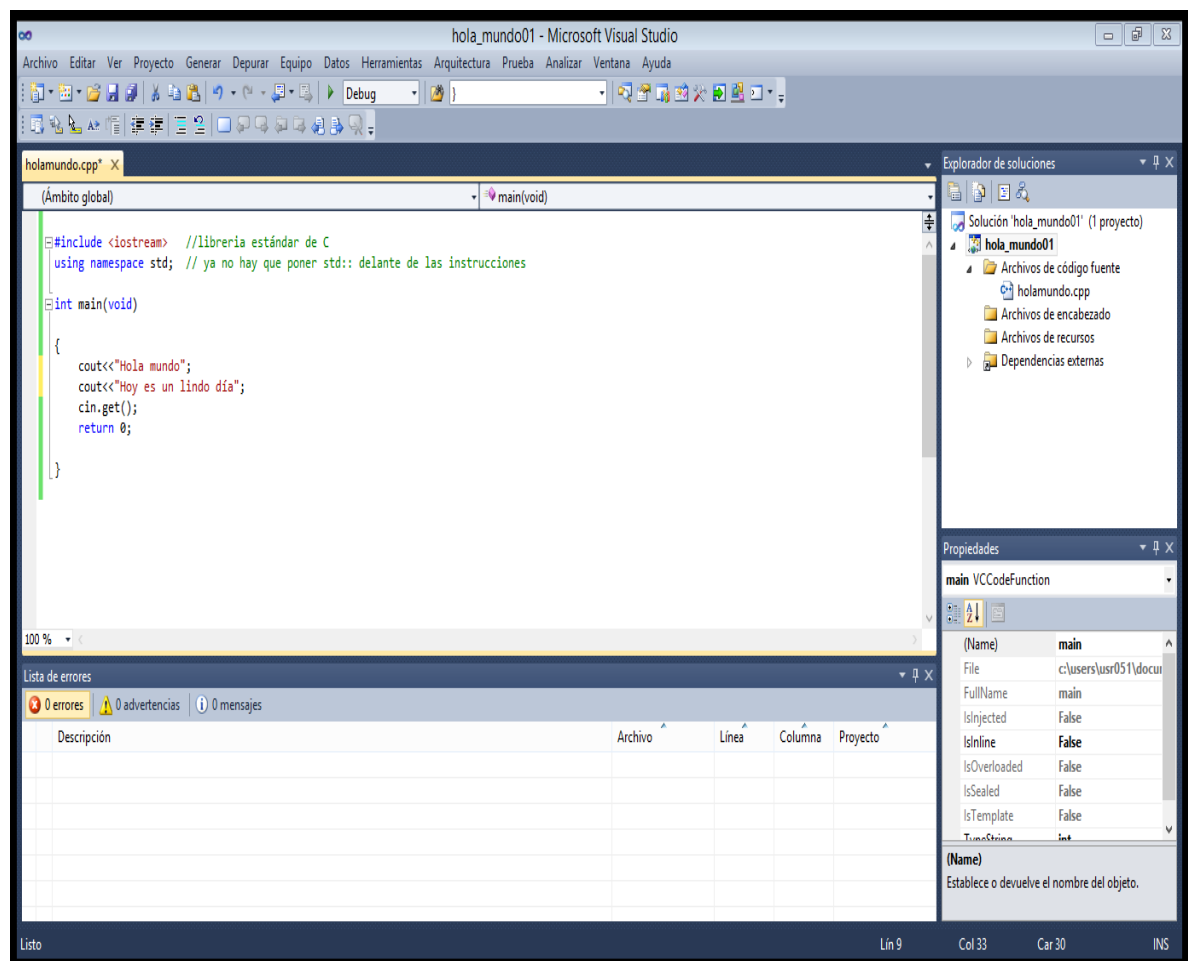


Otros aspectos a considerar:

- Supongamos que se quiere escribir otro mensaje, en cuyo caso deberíamos modificar nuestro código, por ejemplo, si queremos escribir: “Hoy es un lindo día”, entonces escribiríamos lo siguiente:

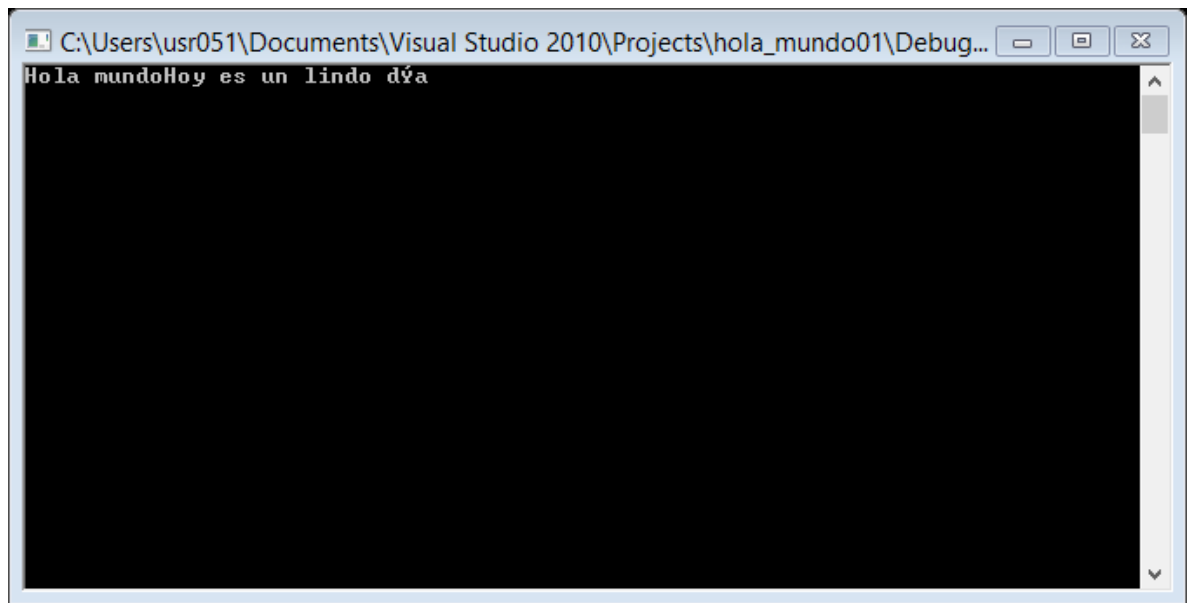
```
cout<<"Hoy es un lindo día";
```

- Quedando de la siguiente forma el código de la aplicación:



- Al correr la aplicación, aparecería lo siguiente:

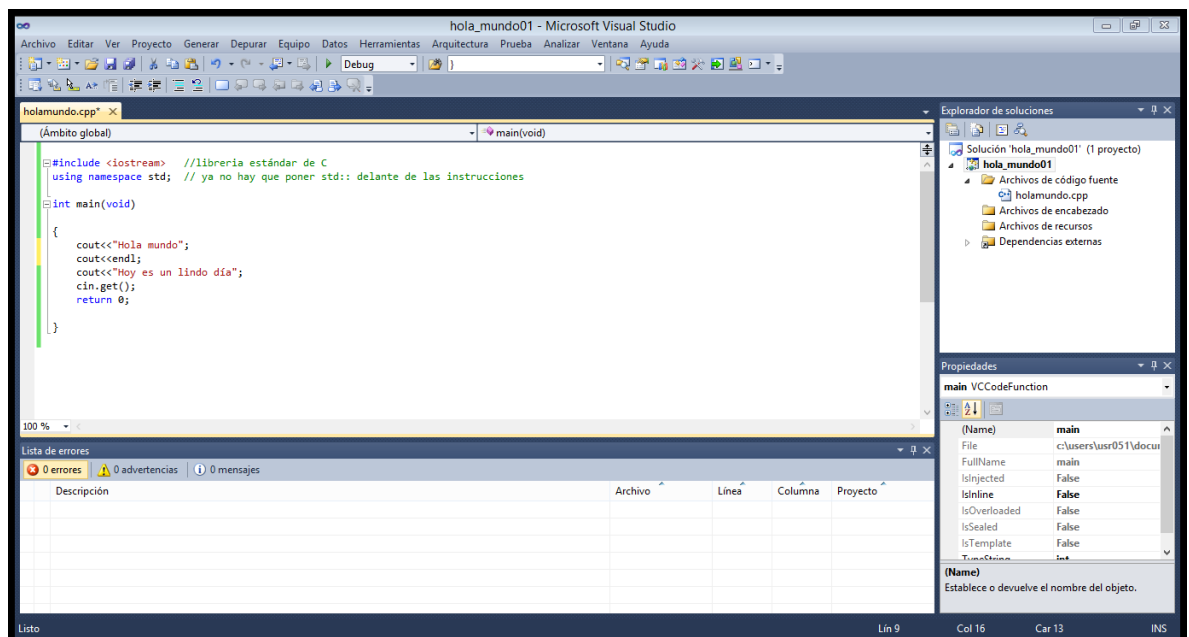
Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



- Como pueden ver, aparecer junto al mensaje anterior, esto no queda bien, pues no se lee cómodamente. Para solucionarlo, podemos utilizar un comando, que baja a la siguiente línea, por lo que podemos utilizar la siguiente instrucción:

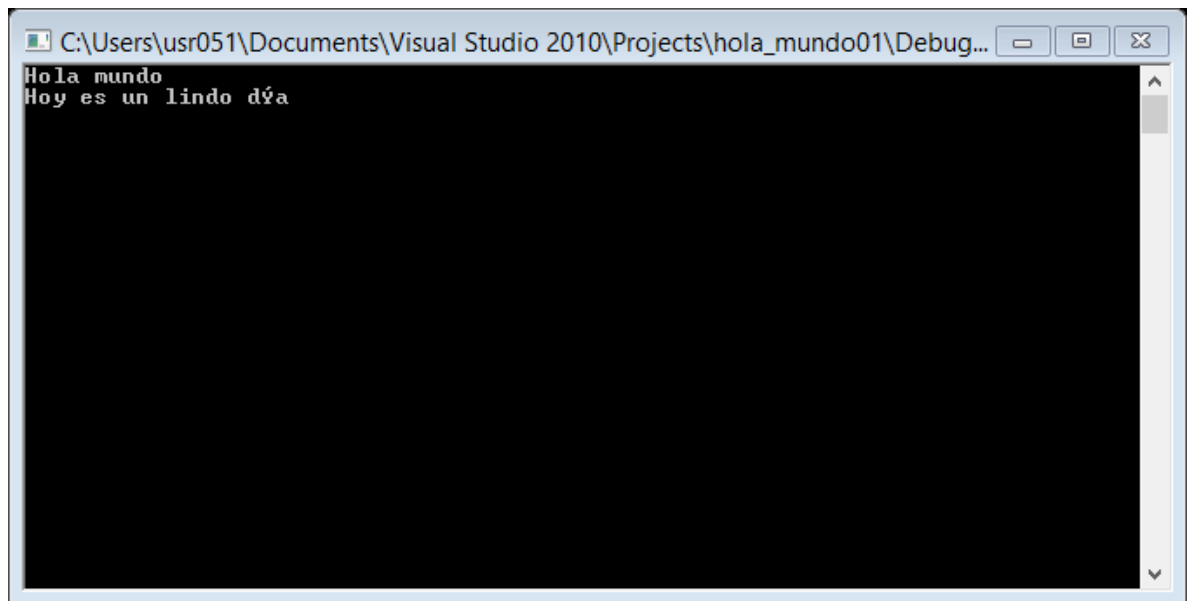
- `cout<<endl;`

- Quedando el código de la siguiente manera:



Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

- Al correr nuevamente la aplicación, se obtendría lo siguiente:



Esto que hemos hecho, se puede realizar de otra forma: Esta forma es visual, creando un proyecto para Windows. O También Utilizando el enfoque Orientado a Objetos.

3.- Utilizando Zinjal, realice el primer ejercicio, sobre las operaciones matemáticas.

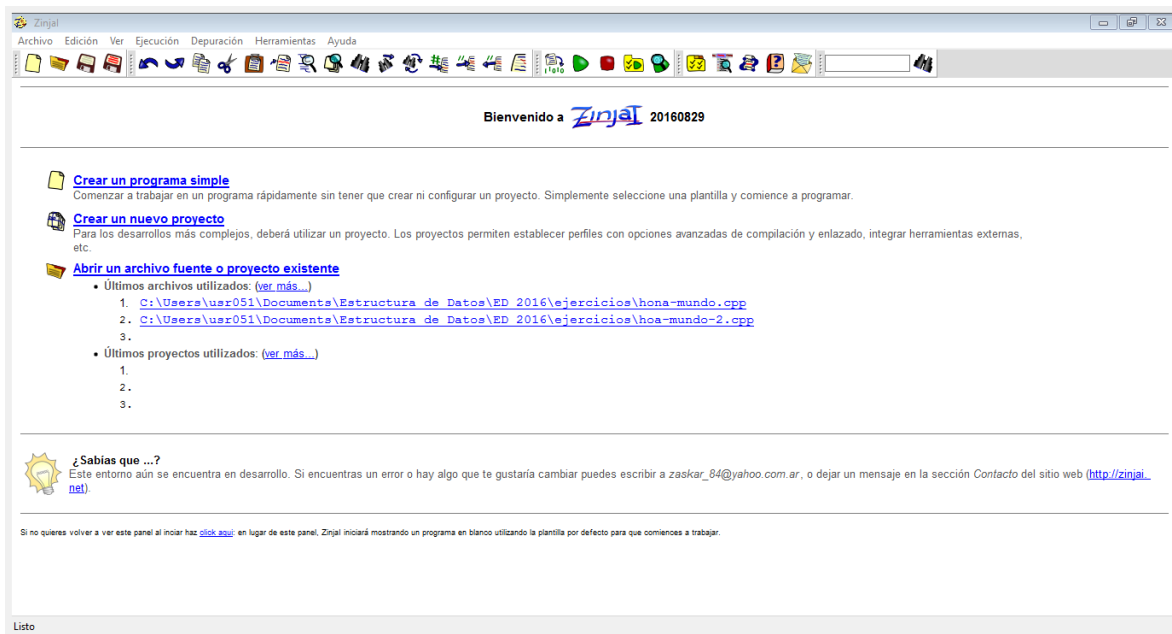
Para comenzar a trabajar en la asignatura, es necesario trabajar con el lenguaje C++, en este caso se utilizará el compilador ZINJAL.

La aplicación a realizar permitirá ingresar dos números y sobre ellos realizar las operaciones básicas de suma, resta, multiplicación y división.

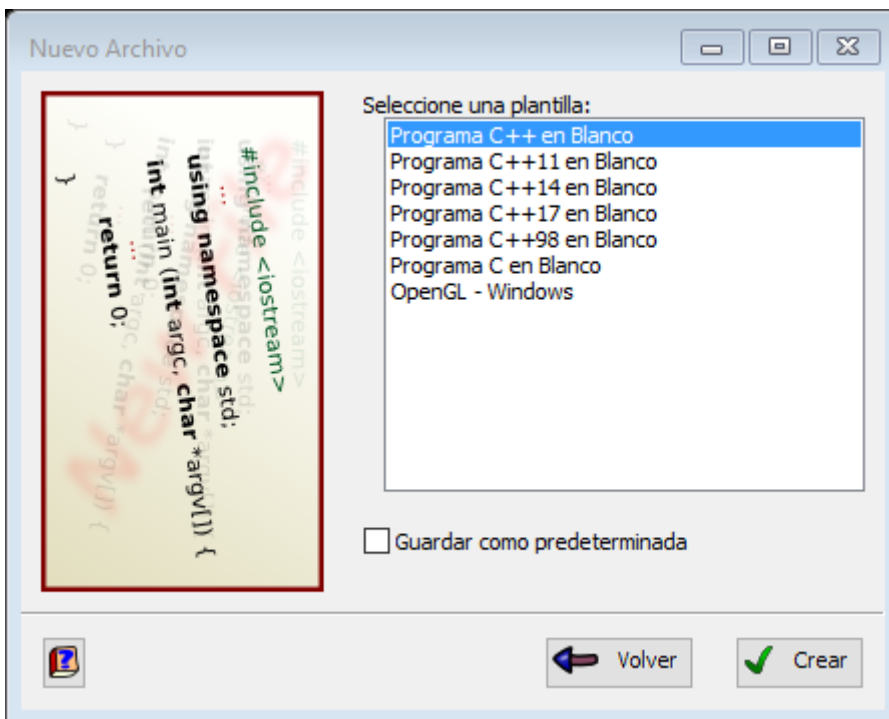
Para comenzar es necesario hacer lo siguiente:

- 1.- Abrir el compilador ZINJAL, aparecerá la siguiente ventana:

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

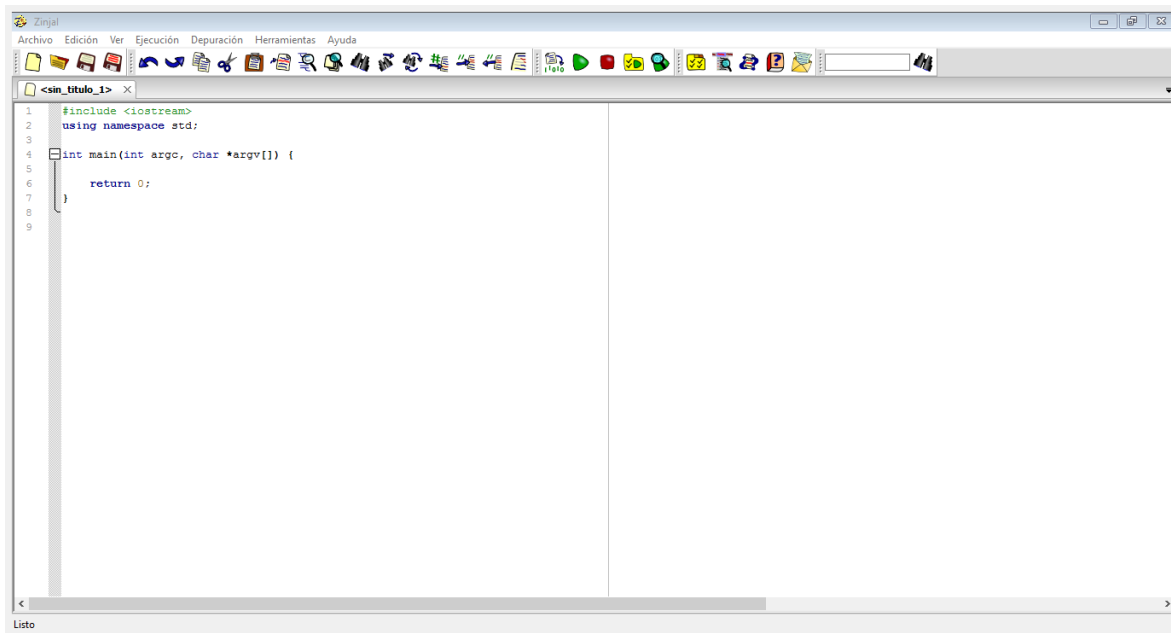


2.- Ir a Crear un programa simple, una vez hecho esto aparecerá el siguiente cuadro de dialogo, donde podrá seleccionar un programa c++ en blanco.



Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

3.- Aparecerá la siguiente interfaz:

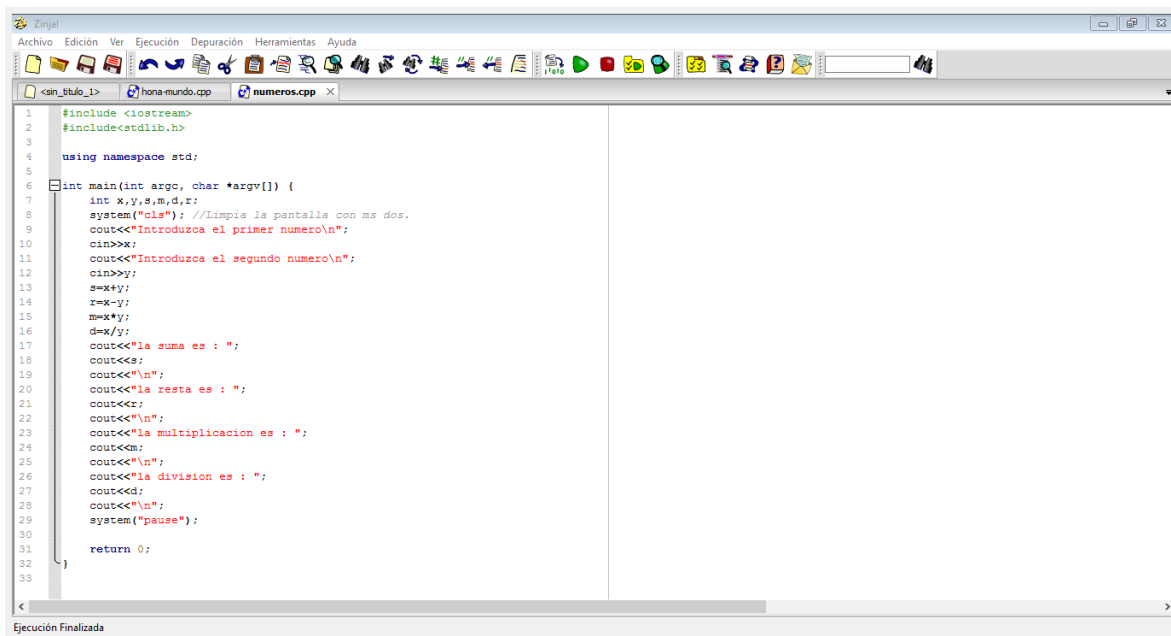


The screenshot shows the Zinjal IDE interface. The menu bar includes Archivo, Edición, Ver, Ejecución, Depuración, Herramientas, and Ayuda. The toolbar contains various icons for file operations, editing, and execution. The main editor window displays a C++ program skeleton with the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char *argv[]) {
5
6     return 0;
7 }
8
9
```

The status bar at the bottom indicates 'Listo'.

3.- Escribir el programa principal y poner en uso las librerías necesarias para comenzar a trabajar y el código que permitirá realizar las operaciones básicas, quedando de la siguiente manera:



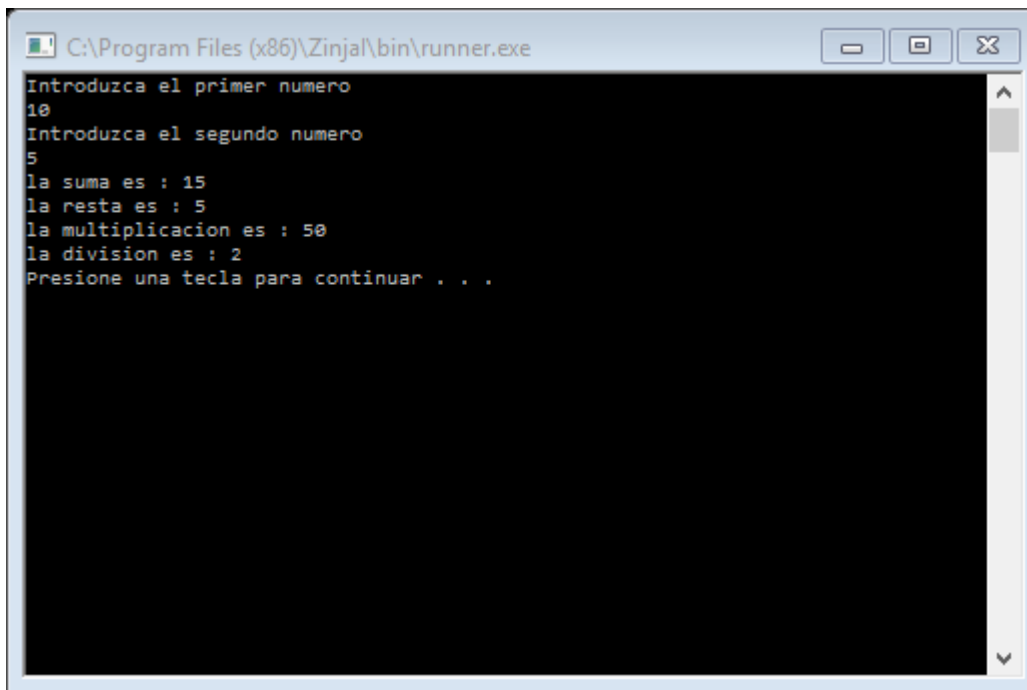
The screenshot shows the Zinjal IDE interface with the file 'numeros.cpp' open. The code implements a program for basic arithmetic operations. The code is as follows:

```
1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 int main(int argc, char *argv[]) {
7     int x,y,s,m,d,r;
8     system("cls"); //Limpia la pantalla con ms dos.
9     cout<<"Introduzca el primer numero\n";
10    cin>>x;
11    cout<<"Introduzca el segundo numero\n";
12    cin>>y;
13    s=x+y;
14    r=x-y;
15    m=x*y;
16    d=x/y;
17    cout<<"la suma es : ";
18    cout<<s;
19    cout<<"\n";
20    cout<<"la resta es : ";
21    cout<<r;
22    cout<<"\n";
23    cout<<"la multiplicacion es : ";
24    cout<<m;
25    cout<<"\n";
26    cout<<"la division es : ";
27    cout<<d;
28    cout<<"\n";
29    system("pause");
30
31    return 0;
32 }
33
```

The status bar at the bottom indicates 'Ejecución Finalizada'.

5.- Guardar y Correr la aplicación y ver el resultado de la misma.

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.



A screenshot of a Windows application window titled "C:\Program Files (x86)\Zinjal\bin\runner.exe". The window contains a text-based interface for a calculator. The text is as follows:

```
Introduzca el primer numero
10
Introduzca el segundo numero
5
la suma es : 15
la resta es : 5
la multiplicacion es : 50
la division es : 2
Presione una tecla para continuar . . .
```

4.- Utilizando Zinjal, realice el segundo ejercicio, sobre HOLA MUNDO

La aplicación a realizar permitirá mostrar un mensaje en la pantalla que dice HOLA MUNDO.

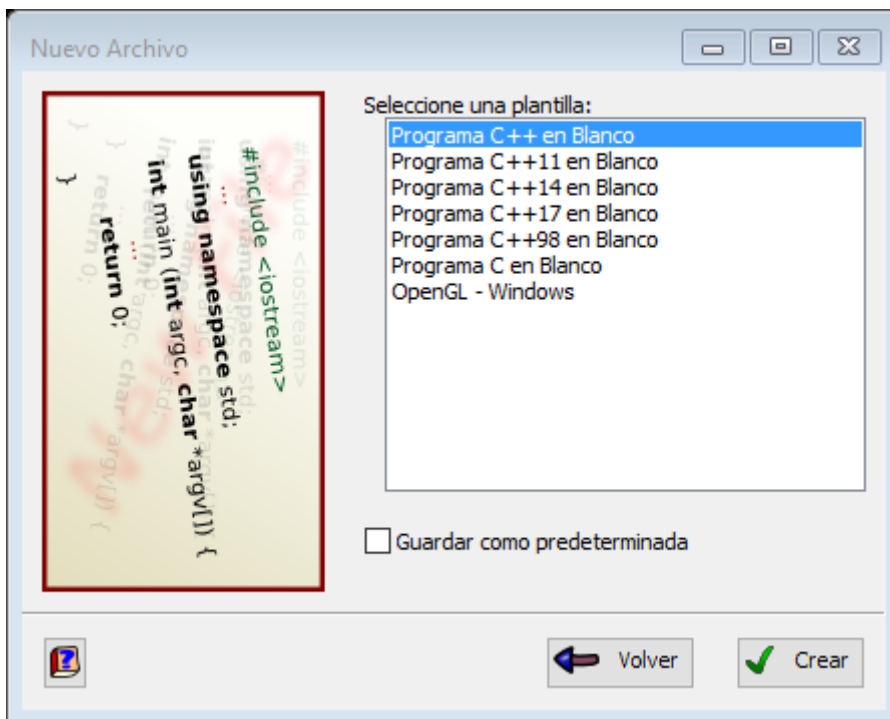
Para comenzar es necesario hacer lo siguiente:

- 1.- Abrir el compilador ZINJAL, aparecerá la siguiente ventana:

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

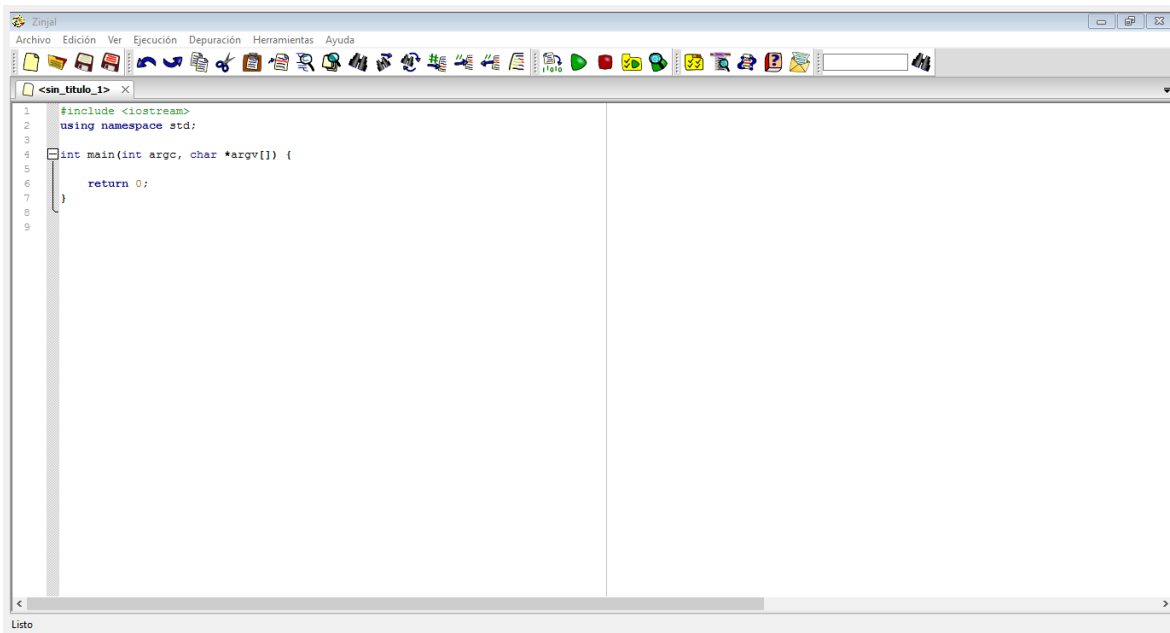


2.- Ir a Crear un programa simple, una vez hecho esto aparecerá el siguiente cuadro de dialogo, donde podrá seleccionar un programa c++ en blanco.

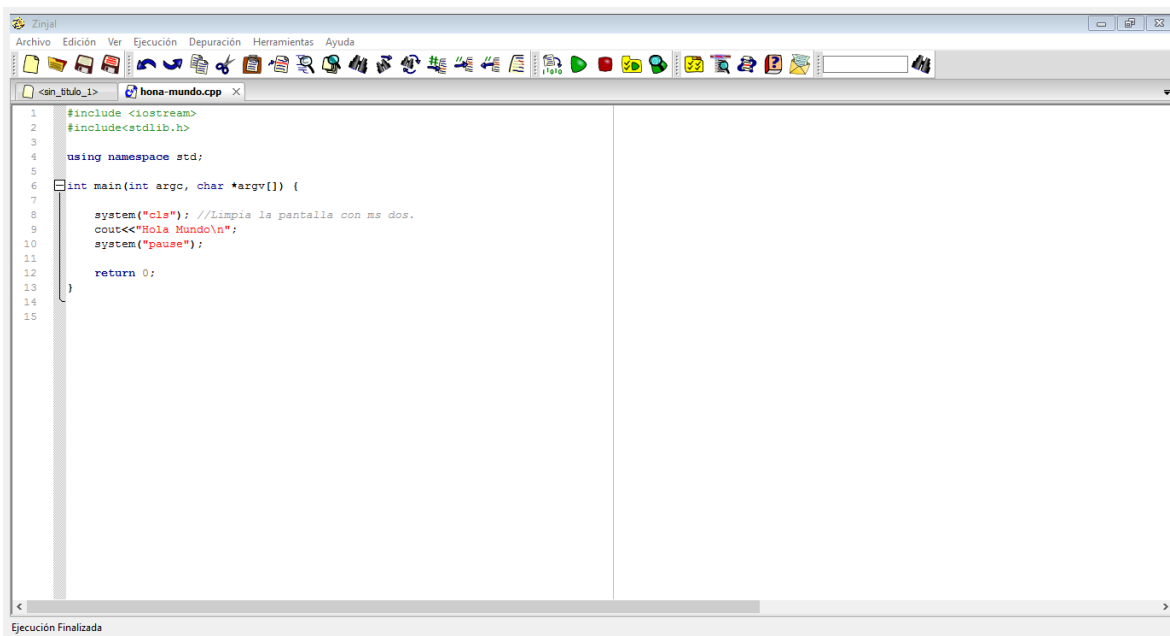


Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

3.- Aparecerá la siguiente interfaz:



3.- Escribir el programa principal y poner en uso las librerías necesarias para comenzar a trabajar y escribir el código necesario, quedando la ventana de la siguiente manera:



5.- Guardar y Correr la aplicación y ver el resultado de la misma.

Este Documento es de uso personal e intransferible, no se permite la reproducción total o parcial de este documento por ningún medio.

