

Programación I

Unidad 1

Unidad E.I.D.

En el desarrollo de esta unidad abordaremos:

- Entorno de trabajo E.I.D. (Entorno integrado de Desarrollo)
- Qué es una solución, que es un proyecto. Tipos de proyectos.
- Características del entorno. Compilación de programas
- Depuración, utilización del entorno Visual Studio .NET para la ejecución de programas.
- Creación de un programa en lenguaje C++.
- Creación de funciones en lenguaje C++.
- Utilización de arreglos en lenguaje C++.

Objetivos

Pretendemos que al finalizar de estudiar esta unidad el alumno logre:

- Poder realizar el primer programa en lenguaje C++ desde su escritura hasta su ejecución, conociendo las etapas intermedias.
- Tener una primera aproximación a un entorno de trabajo.
- Crear y utilizar funciones en lenguaje C++.
- Crear y utilizar arreglos en lenguaje C++.

Aclaraciones previas al estudio

En esta unidad, el alumno encontrará:

- Contenidos
- Conceptualizaciones centrales
- Actividades



Usted debe tener presente que los contenidos presentados en esta unidad no ahondan profundamente en el tema, sino que pretenden ser un recurso motivador para que, a través de la lectura del material, la bibliografía sugerida y el desarrollo de las actividades propuestas alcance los objetivos planteados. Cada unidad constituye una guía cuya finalidad es facilitar su aprendizaje.

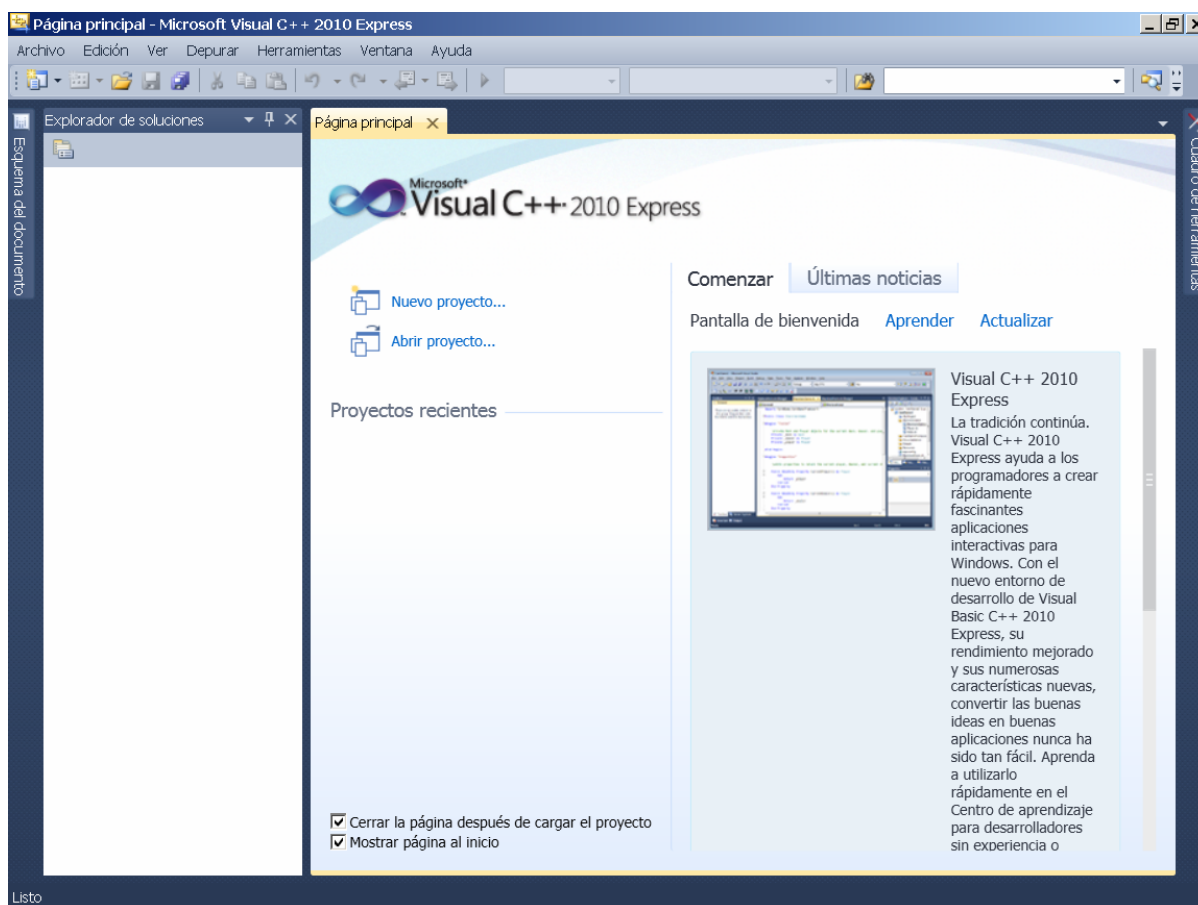
EL ENTORNO INTEGRADO DE DESARROLLO

El EID, Entorno Integrado de Desarrollo (*IDE, Integrated Development Environment*), es un entorno de programación que proporciona todas las herramientas que se necesitan para desarrollar programas de aplicación en C / C++.

Uno de los entornos que se pueden utilizar para estudiar C y C++ es Microsoft Visual C++. Está disponible en dos opciones. Microsoft Visual Studio 2010 es un entorno comercial en varios idiomas que admite la programación en C++ y otros lenguajes (C#, Visual Basic). Se incluye Microsoft Visual C++ para programar en C y C++. Para que sea más fácil de aprender programación en C++, Microsoft creó Microsoft Visual C++ 2010 Express. Se trata de un entorno de programación completamente funcional que soporta tanto C como C++ (también apoya a otro hijo del lenguaje C, llamado C++/CLI). Podemos usar Microsoft Visual Studio 2010 o Visual C++ 2010 Express para crear aplicaciones C y C++. Microsoft Visual C++ 2010 Express es una aplicación gratuita. Para utilizarlo, deberemos descargarlo desde el sitio web de Microsoft. Una vez en el sitio, busque y haga clic en el enlace de Visual Studio Express. A continuación, busque y haga clic en Microsoft Visual C++ 2010 Express. Después de instalarlo deberá registrarse en el sitio de Microsoft ya que la instalación es válida por 30 días.

Para arrancar Microsoft Visual C++ 2010 Express, vamos a:

Inicio -> Programas -> Microsoft Visual Studio 2010 Express -> Microsoft Visual C++ 2010 Express:



El EID posee un conjunto importante de herramientas:

- Un editor con características propias de tratamiento de textos tales como « buscar y reemplazar » o « cortar y pegar ».
- Un sistema de gestión de soluciones (administración de archivos), que le facilita la grabación y recuperación de sus archivos fuente.
- Un compilador, que produce programas EXE autónomos que se pueden ejecutar directamente desde una ventana de consola.
- Un sistema de ayuda de referencias cruzadas denominado Help de Contexto (intellisense), que proporciona información sobre el lenguaje y el entorno.
- Un depurador de errores (Debugger)
- Un linkeador.
- La posibilidad de ejecutar nuestro programa dentro de la misma interfaz

El menú principal

La sección superior de Visual Express C++ mostrará el menú principal dividido en categorías como Archivo o Edición. El menú principal se usa como el menú de una aplicación normal.

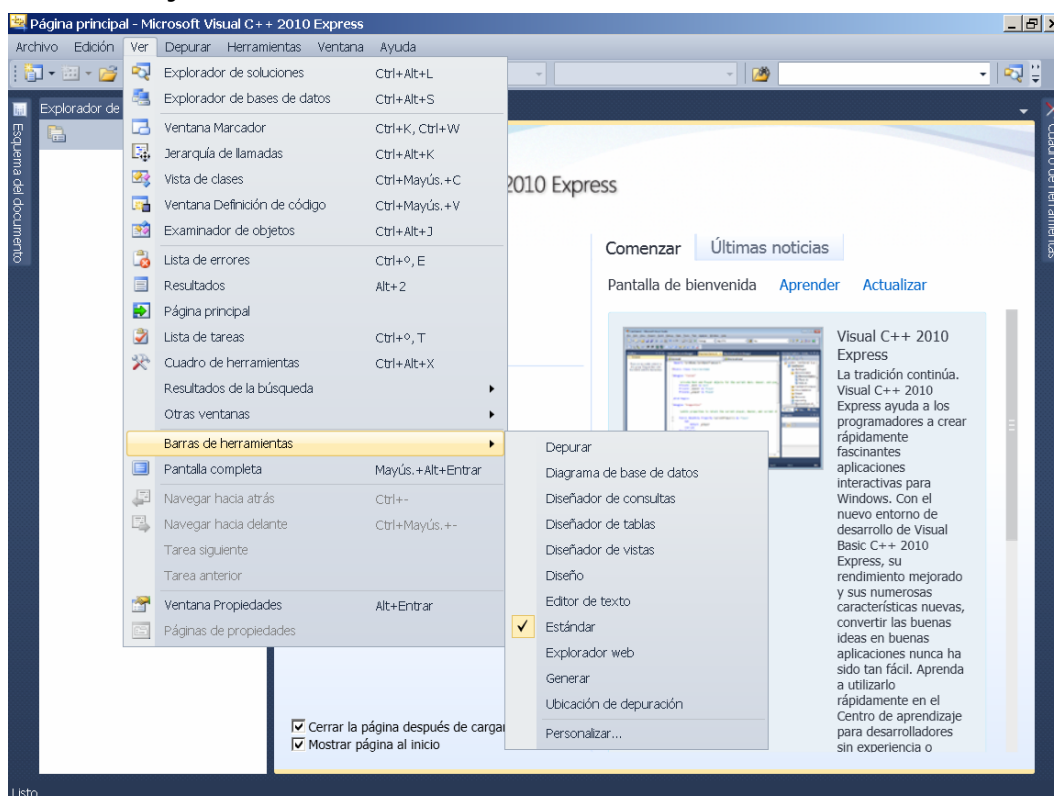
Las barras de herramientas

En el menú principal, hay una barra de herramientas. Normalmente, cuando el VSEXPRESSC++ está corriendo, se muestra la barra de herramientas estándar por defecto. Algunas barras de herramientas adicionales se mostrarán en respuesta a alguna acción de nuestra parte. Sin embargo, podemos mostrar otras barras de herramientas si queremos.

En el menú principal, hacemos clic en

Ver -> Barras de herramientas y clic en la barra de herramientas de su elección

- Clic en el menú principal o barra de herramientas. clic en la barra de herramientas deseada



La página de inicio

La página de inicio es el parte principal de Visual C++ Express cuando lo ejecutamos. Se inicia en la parte superior con una pestaña denominada página Principal.

La ventana de la aplicación muestra Microsoft Visual C++ 2010 Express. La sección de la derecha muestra dos etiquetas: **Nuevo proyecto...** y el **Abrir proyecto...** Si ya creamos o abrimos algunos proyectos, sus nombres aparecen en **Abrir proyecto...**

Cuando se trabaja en un proyecto, la página de inicio suele estar en el fondo, y está representada por una lengüeta. Para mostrar la página de inicio en cualquier momento puede:

- Hacer clic en la lengüeta de página Principal en la sección superior del Editor de código
- En el menú principal, puede hacer clic en Ver -> Página Principal

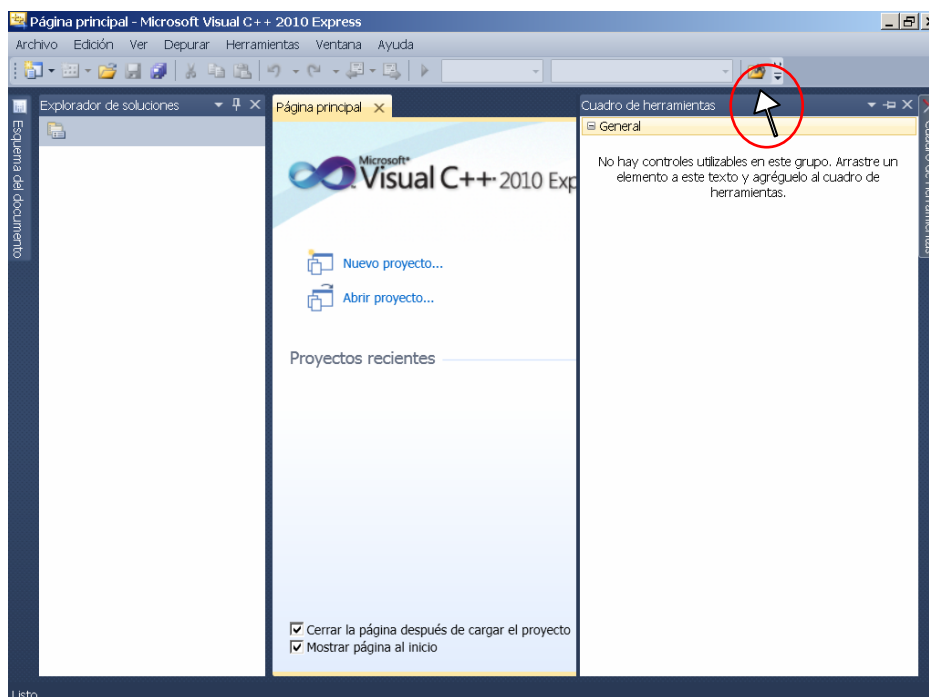
Mostrar y cerrar una ventana

Al crear o abrir un proyecto, Visual C++ Express abre algunas ventanas disponibles. Estas son las ventanas más utilizadas. Si alguna de ellas no se van a utilizar regularmente en los proyectos, podemos eliminarla de la pantalla. Para ocultar una ventana, puede hacer clic en el botón Cerrar.

Todas las ventanas que se pueden utilizar están listadas en el menú Ver. Por lo tanto, si una ventana no se ve, podemos hacer clic en Ver en el menú principal y hacer clic en una ventana a elección.

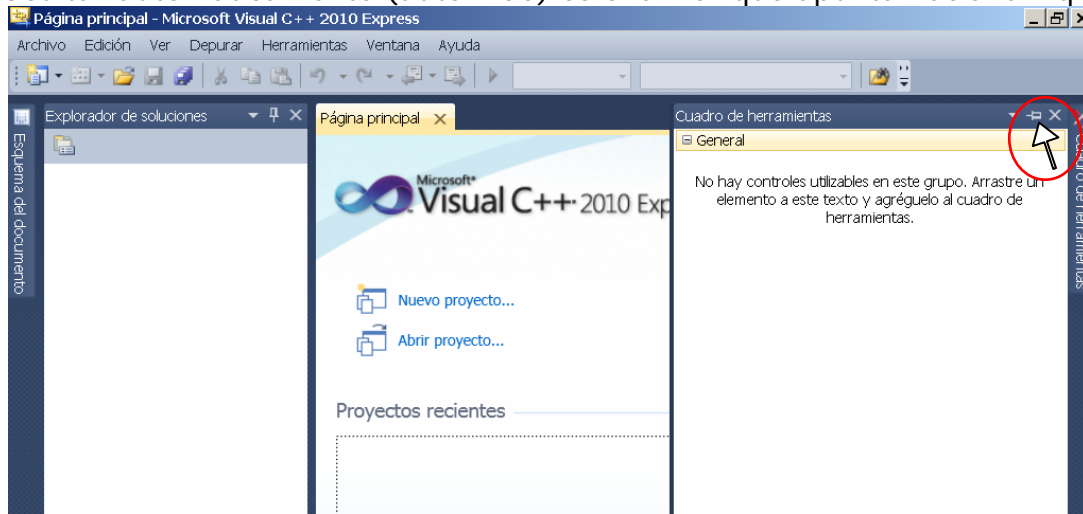
Auto Ocultar una ventana

Cuando se trabaja en un proyecto, se utilizaran un conjunto de ventanas, cada una con un propósito específico. Algunas ventanas están representadas por un icono y esconden el resto del cuerpo. Para mostrar esa ventana, puede colocar el mouse sobre ella. Esto ampliaría la ventana:

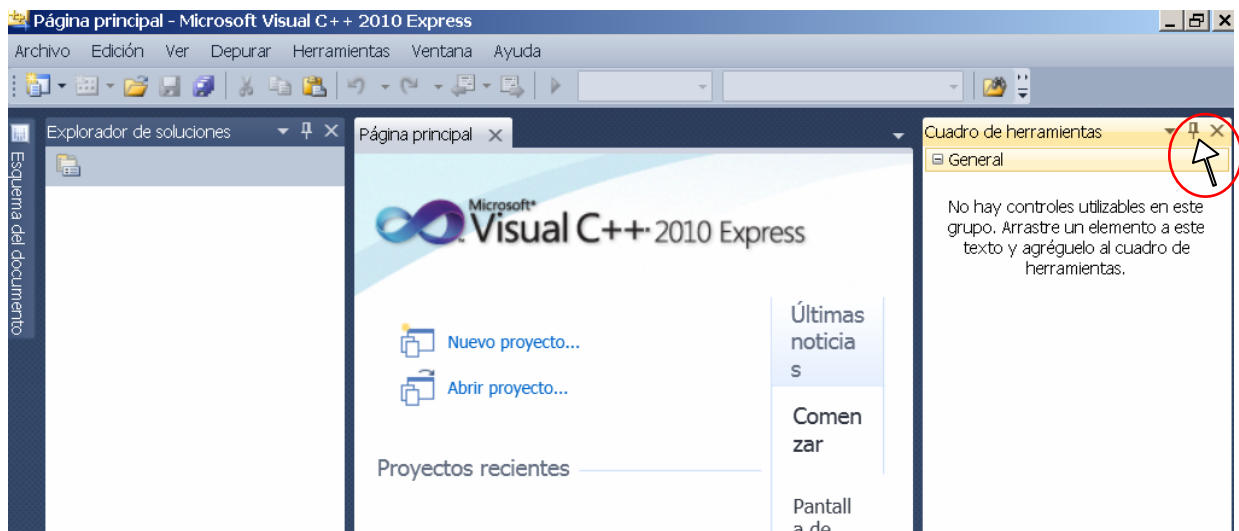


Si expande una ventana, se mostrará una barra de título con tres botones. Uno de los botones se llama

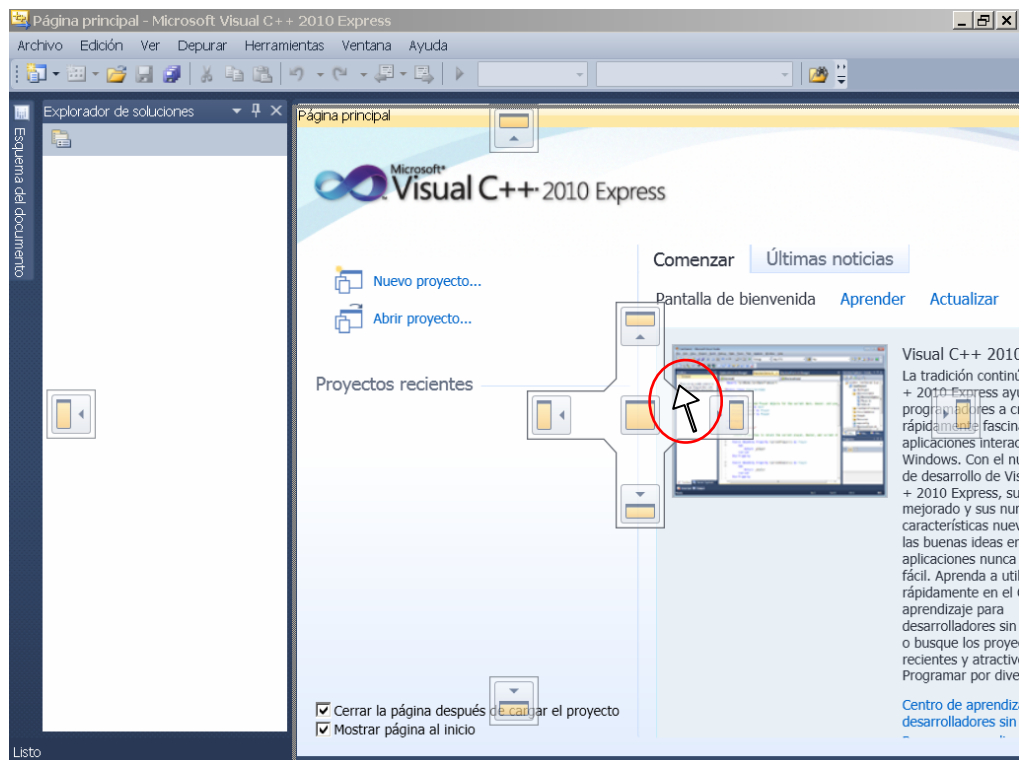
Ocultar automáticamente (auto hide) es el alfiler que apunta hacia la izq.:



Si expandimos una ventana, y luego decidimos que no la necesitamos más, movemos el mouse fuera de ella. La ventana vuelve a su estado anterior. Sobre la base de esta funcionalidad, si estamos trabajando con una ventana y movemos el mouse afuera, esta se contrae. Si la necesitamos nuevamente, tendríamos que volver a abrirla de la misma forma. Si vamos a trabajar con una ventana por un tiempo prolongado, podemos mantenerla abierta incluso si movemos el mouse fuera de ella. Para ello, hacemos clic en el botón Ocultar automáticamente y veremos al icono que apuntaba a la izquierda apuntar hacia abajo:



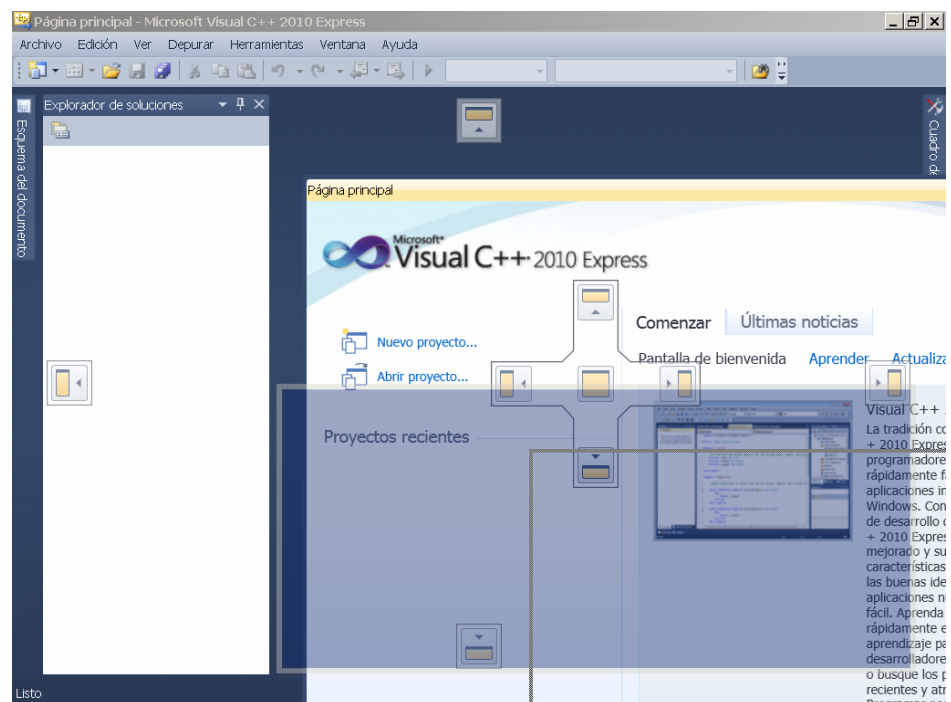
De forma predeterminada, el VExpressC++ instala algunas ventanas a la izquierda y otras a la derecha de la pantalla. Podemos cambiar este formato si queremos. Para ello, expandimos la ventana, y a continuación, hacemos clic en su barra de título y empezamos a arrastrarla. Mientras se está arrastrando a una nueva posición, aparecen cinco cajas:



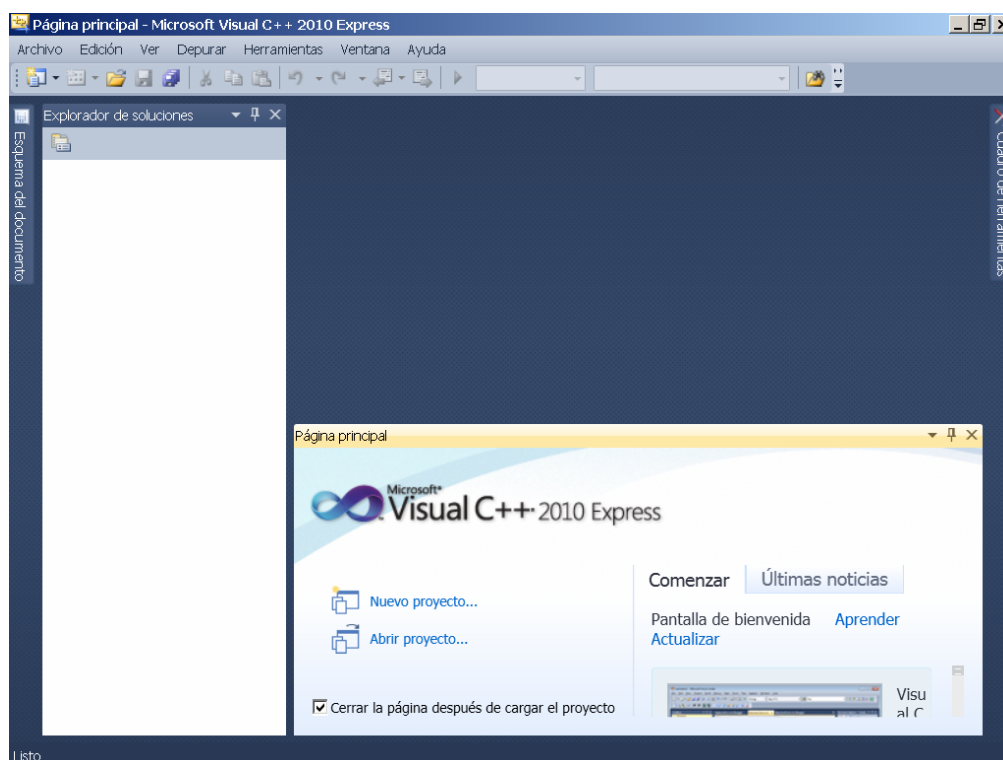
Acoplamiento de una ventana en la pantalla

Para colocar una ventana en un lado específico, arrastramos su barra de título para el cuadro que representa a ese lado. Al llegar a un lado, un rectángulo azul transparente aparecerá para indicar la posición posible. Soltamos entonces la ventana ya sea a la izquierda de la caja para colocarla a la izquierda de la pantalla, o en la caja más a la derecha de la guía para colocarla a la derecha de la pantalla o abajo.

Aquí un ejemplo:



En este elegimos la del abajo, entonces ventanas estarán de forma:



caso
caja
las
esta

Preparación del EID

En este curso, vamos a aprender el lenguaje C++, que es el que utiliza VExpressC++. No vamos a aprender a crear aplicaciones gráficas. En cambio, las aplicaciones que vamos a realizar mostrarán sus resultados en una ventana negra llamada ventana de comandos o el símbolo del sistema. Estas se denominan aplicaciones de consola, ya que tradicionalmente se ejecuta en la consola de comandos (en Windows XP será cmd).

Soluciones, Proyectos y Archivos

Las aplicaciones creadas en VExpressC++ 2010 se componen de soluciones, proyectos y archivos.

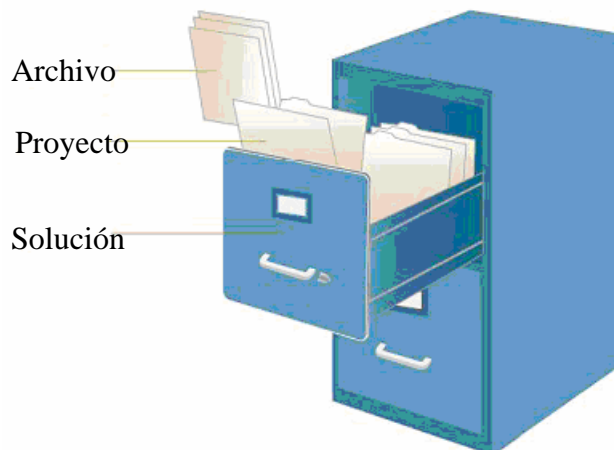
Definición

Una solución es un contenedor que almacena los proyectos y archivos para una aplicación completa.

Un proyecto también es un contenedor, pero que almacena los archivos asociados a ese proyecto en particular.

Aunque la idea de soluciones, proyectos y archivos puede sonar confuso, el concepto de colocar las cosas en los contenedores no es algo nuevo. Pensemos en una solución como algo similar a un cajón de un archivador. Un proyecto a continuación, es similar a una

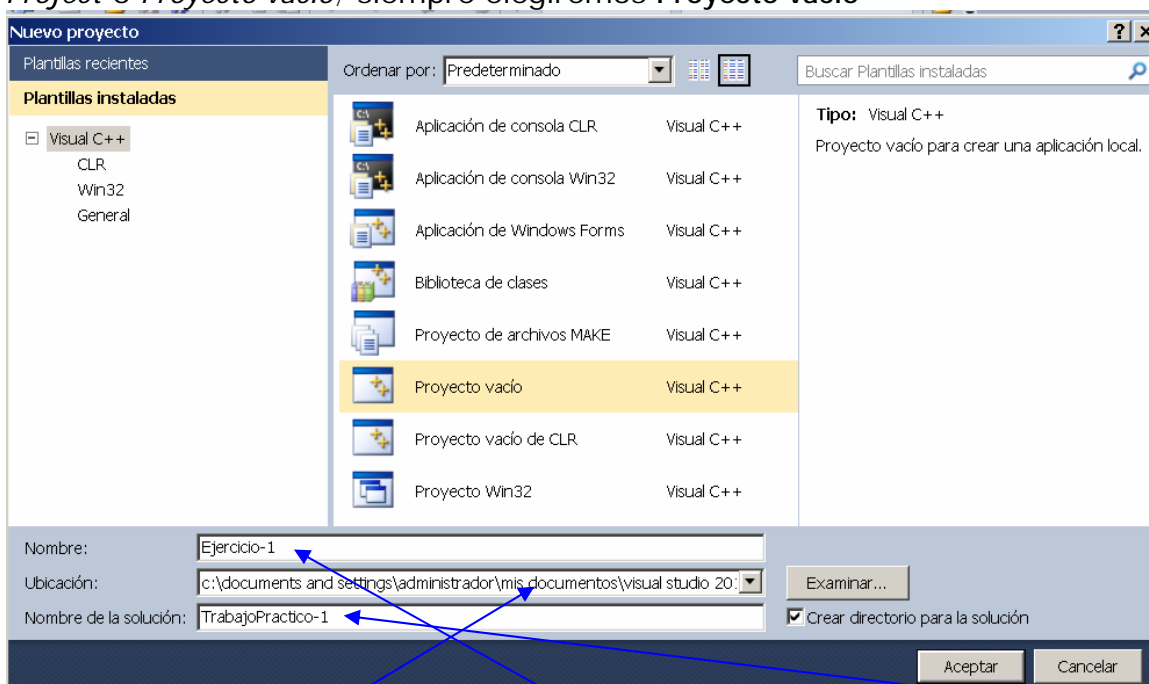
carpeta de archivos que usted guarda en el cajón, y un archivo es similar a un documento que se almacenan en la carpeta de archivos. Podemos colocar varias carpetas de archivos en un cajón archivador, al igual que podemos poner muchos proyectos en una solución. También se puede almacenar muchos documentos en una carpeta de archivos, similar a la manera que podemos almacenar muchos archivos en un proyecto.



A continuación desarrollaremos paso a paso nuestro primer ejercicio. Para crear una aplicación de consola, en VExpressC++ podemos:

- En la página de inicio, hacer clic en **Nuevo proyecto**
- En el menú principal, hacer clic en **Archivo -> Nuevo proyecto...**
- En la barra de herramientas, hacer clic en el botón **Nuevo proyecto**
- Pulsar las teclas **Ctrl + N**

Cualquiera de estas acciones mostrará el cuadro de diálogo Nuevo proyecto. Si bien podemos hacer clic en cualquiera de las opciones: *Aplicación de consola Win32*, *Win32 Project* o *Proyecto vacío*; siempre elegiremos **Proyecto vacío**



En nombre del Proyecto escribiremos **Ejercicio-1** y en el nombre de la Solución **TrabajoPractico-1**

Le indicaremos al Entorno donde ir a buscar o grabar cada componente. En nuestro caso nos deberá quedar de la siguiente manera.

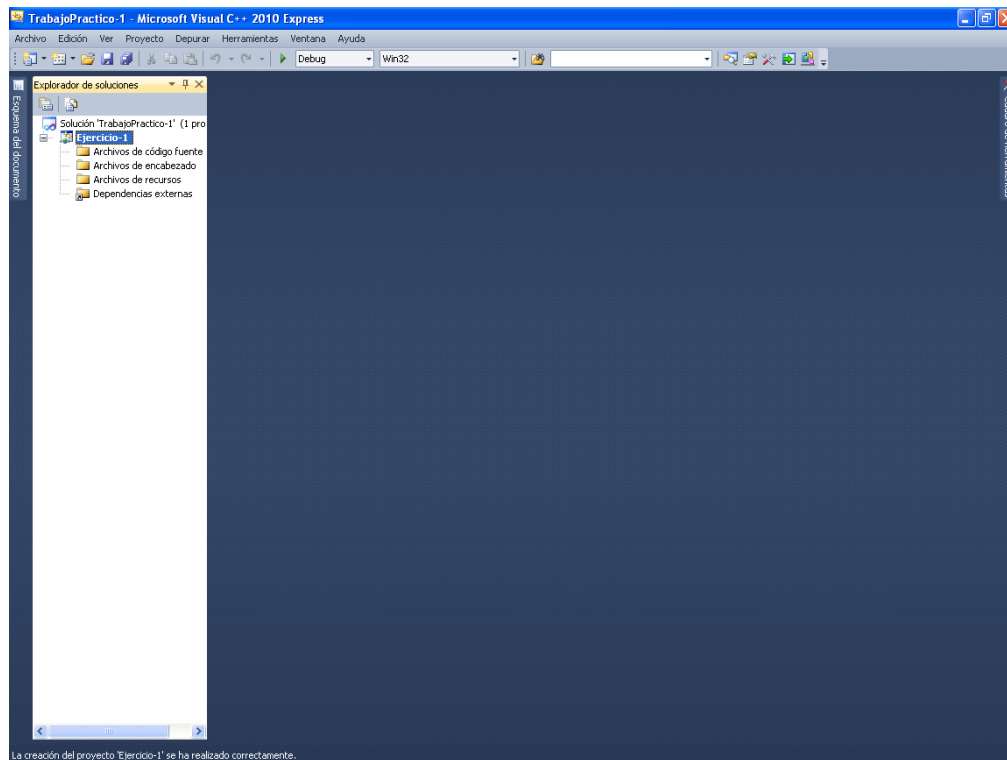
Veamos brevemente cada uno de los NOMBRES ELEJIDOS:

- Nombre de la solución: **TrabajoPractico-1**
Especifica el nombre de la solución que para este curso deberá ser el nombre del trabajo práctico a realizar.
- Nombre del proyecto: **Ejercicio-1**
Especifica el nombre del proyecto dentro de la solución.
Utilizaremos los nombres de los ejercicios designados por la cátedra a realizar en el curso. En este caso es el ejercicio N°1 de la practica N°1.
- Directorio de la solución:
Especifica el directorio en el cual se grabará la solución con todos los proyectos y archivos que la compongan.

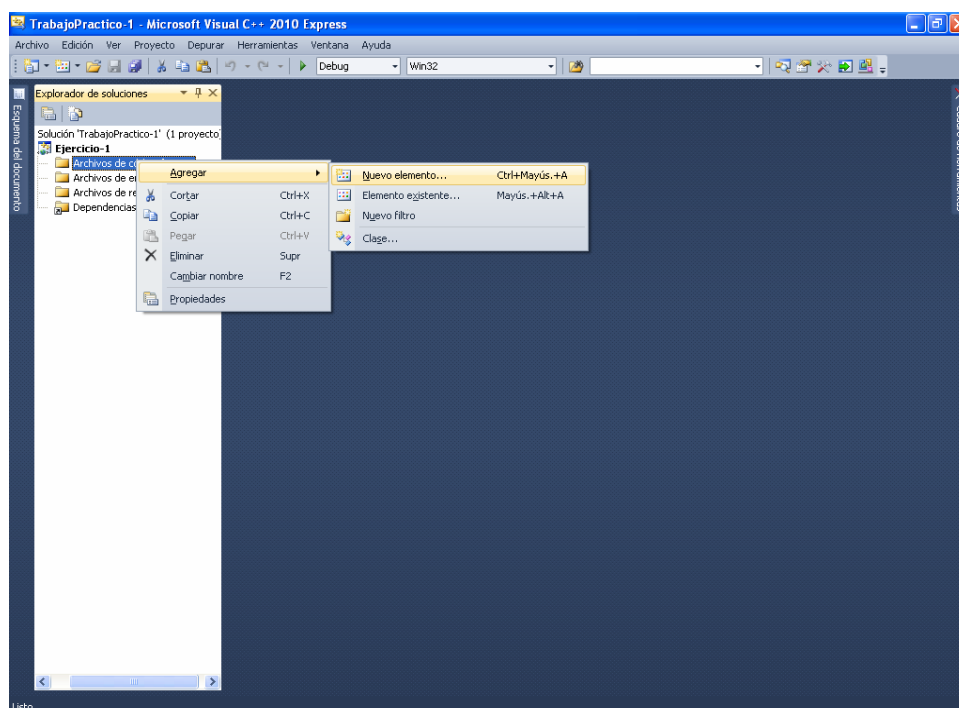
Una vez especificados los nombres de la Solución, del Proyecto y el directorio para los mismos, presionamos **Aceptar**.

Creación de programas: la edición.

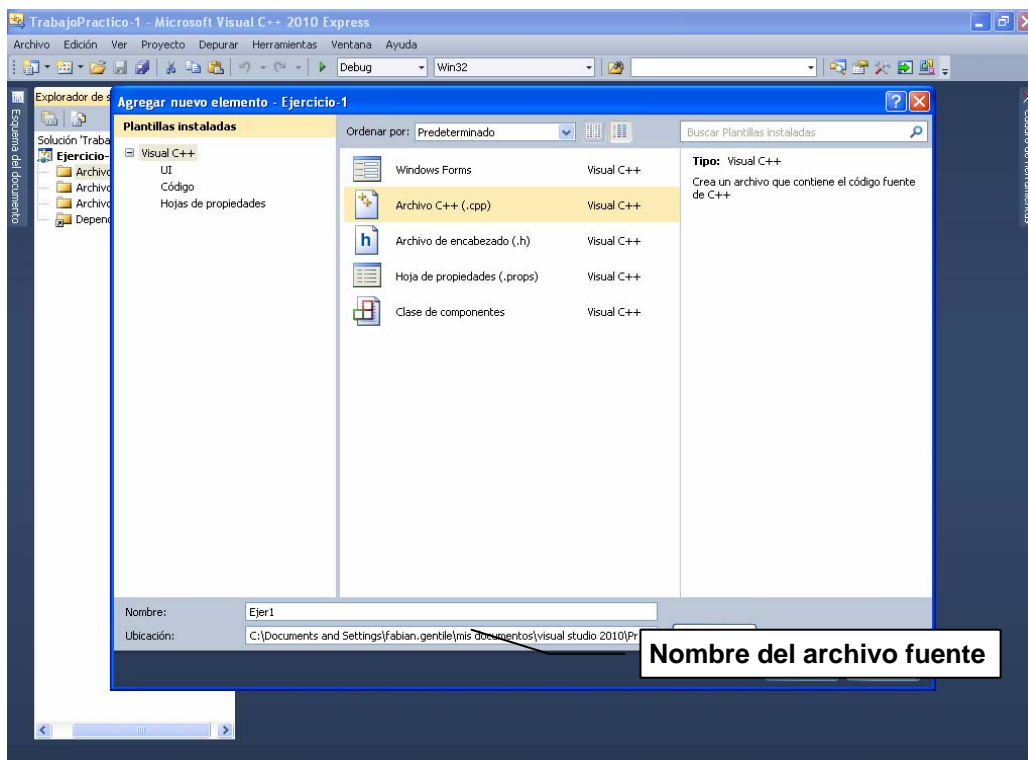
Una vez aceptado aparece a la izquierda la Ventana del Explorador de solución con la estructura de carpetas vacía del proyecto llamado Ejercicio-1 dentro de la Solución llamada TrabajoPractico-1.



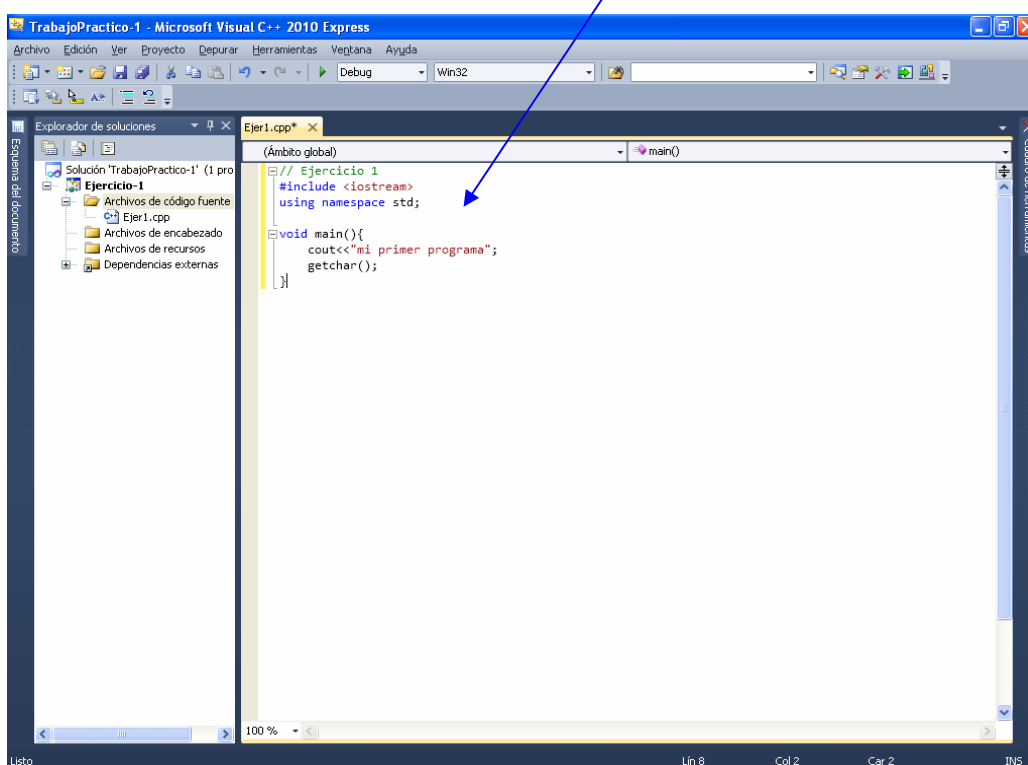
A continuación haremos nuestro primer programa en C++, el cual analizaremos línea por línea más adelante. Lo primero que haremos es crear nuestro programa fuente. Nos paramos en la ventana del explorador y elegimos con el botón derecho sobre la carpeta **Archivos de código fuente- Agregar - Nuevo elemento**



Aparece la siguiente ventana donde elegimos **Archivo C++(.cpp)** y le colocamos el nombre **Ejer1.cpp**. Aceptamos y nos encontramos con el editor, donde podremos transcribir el diagrama de resolución en el programa C/C++.



Se abre entonces una nueva ventana y ahora estamos ya en condiciones de comenzar a escribir el programa. Transcribimos el siguiente código



```
// Ejercicio 1
#include <iostream>
using namespace std;

void main(){
    cout<<" Mi Primer Programa ";
    getch();
}
```



Observe como está escrita la palabra `main()` siempre deberá escribirla de esta manera, ya que si por ejemplo la escribe en mayúsculas (`MAIN`) o la primer letra en mayúsculas y el resto en minúsculas (`Main`), el compilador dará un error.

Grabación de programas en disco.

Si no ha salido del editor, el programa fuente estará todavía en memoria, pero si por cualquier circunstancia se apaga la computadora, podría perderse. Por ello, es una buena idea, antes de seguir, guardarlo o grabarlo en disco. Para hacer esto se debe seleccionar la orden **GUARDAR** del menú **ARCHIVO**.

Normalmente se deben guardar los programas con nombres de archivos específicos y significativos que le recuerden siempre su contenido o función.

Un método alternativo para grabar es utilizar la tecla de función **CTRL-S** en lugar de ejecutar la orden específica **GUARDAR**.

Ahora estamos en condiciones de compilar el programa fuente (Compiler) y ejecutar el programa compilado (**Run**).

Una vez que tenemos nuestro programa fuente terminado, es hora de probarlo. Para poder realizar esto, se lo debe compilar.

Compilación

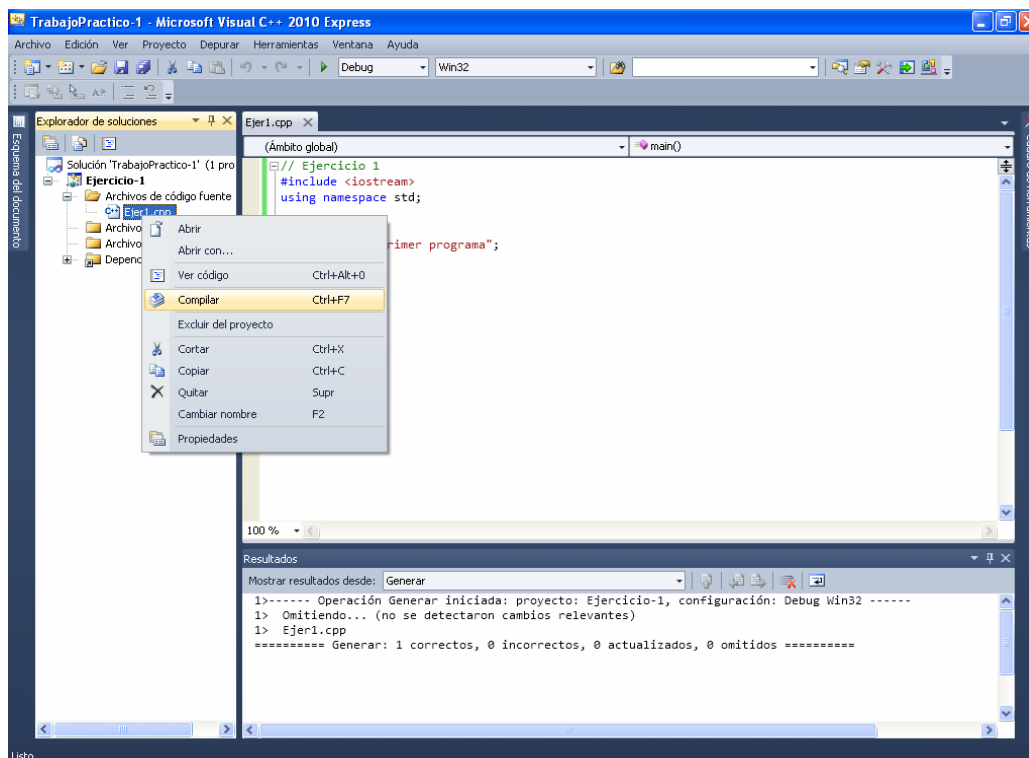
Para compilar existen tres formas. Vamos al menú **Generar** y seleccionamos **Compilar**

Con el botón derecho sobre el archivo fuente **Compilar**

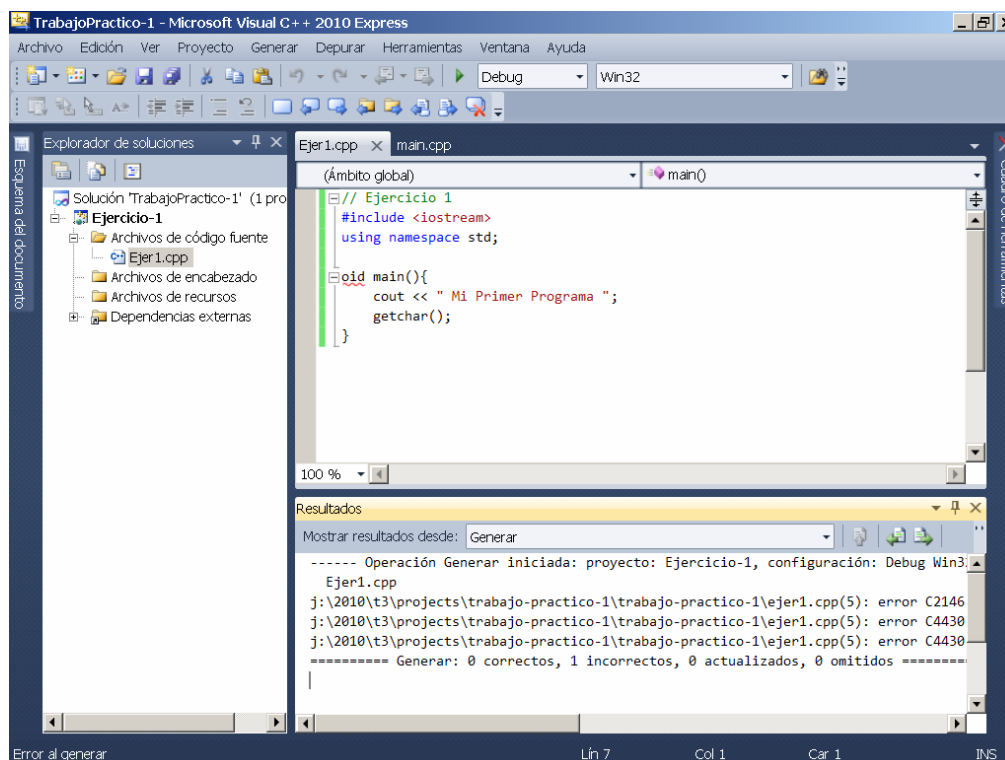
Pulsar **CTRL + F7** (desde cualquier parte del entorno).

Una vez dentro del menú **Generar**, se elige la opción u orden **Compilar**.

La secuencia de teclas que activa esta orden es **CTRL + F7**. Tras ejecutar esta orden se compila el programa, aparece la **Ventana de Resultados** en el fondo y visualiza el proceso de la compilación del programa.

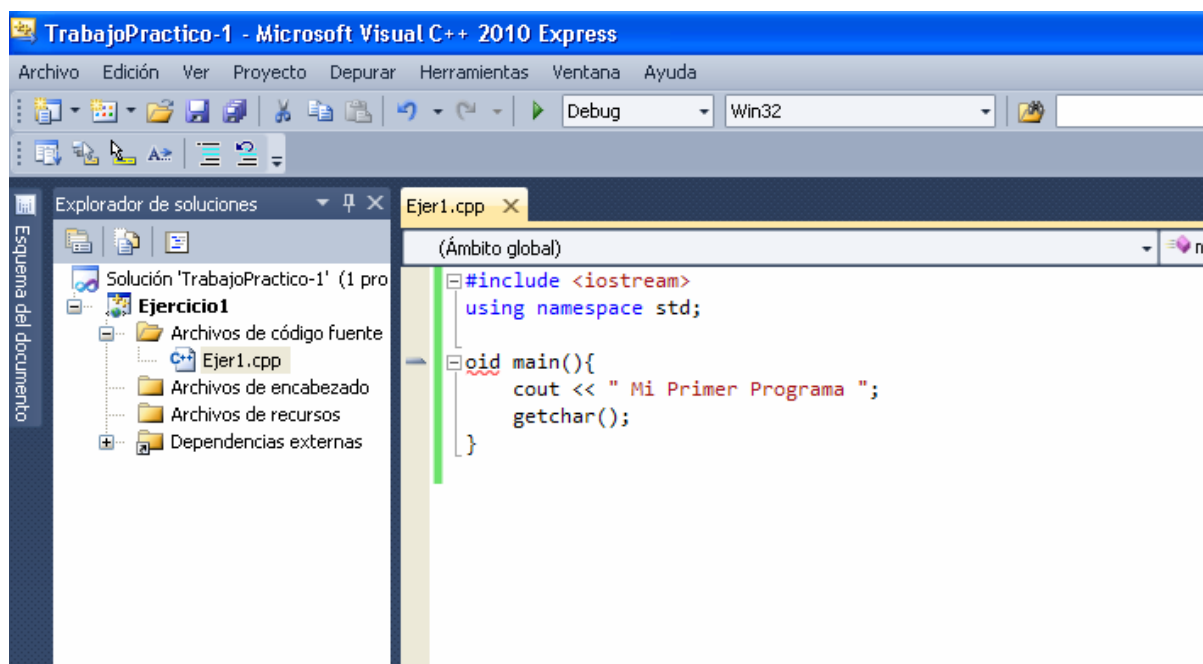


Si ningún error sucede durante la compilación, aparece el mensaje:
Generar: 1 correcto, 0 incorrecto, 0 actualizados, 0 omitidos

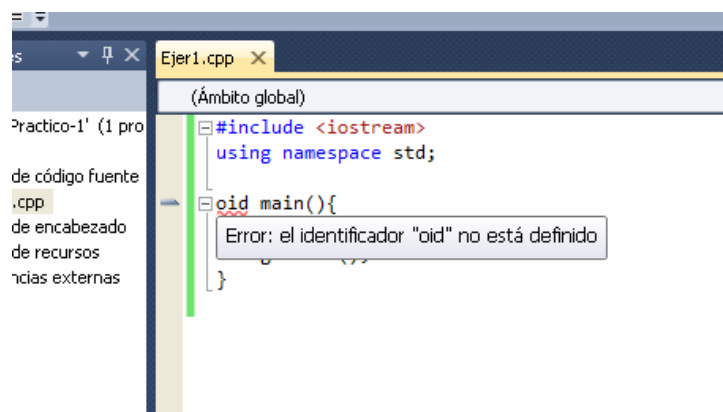


Si ocurre un error durante la compilación aparece el mensaje:
Generar: 0 correcto, 1 incorrecto, 0 actualizados, 0 omitidos

Se puede observar que la línea que contiene el error está marcada:



Al posicionarnos con el Mouse sobre el posible error, nos lo indicara con un mensaje, como se ve en siguiente imagen:



Para corregir el error se debe posicionar el cursor en el sitio correcto, escribir la sentencia como corresponda y nuevamente grabar la versión correcta del programa y volver a compilar. En este caso, cambiar el identificador `oid` por `void` (en minúsculas) solucionará el error.

Ejecución (Run)

Al ejecutar el programa, y dependiendo de la configuración del EID, es posible que nos haga una pregunta indicando la creación de la solución completa. Solo debemos decir que sí, ya que es la forma en la cual se va a realizar la Linkedición antes mencionada y se ejecutará la consola en donde podremos observar la interfaz que el usuario verá de nuestro programa.

Transcribir el siguiente programa, siguiendo las instrucciones planteadas en esta unidad, compilarlo, ejecutarlo y verificar su funcionamiento.

```
/*
Grupo 1
Apellido Nombre, Apellido Nombre,

Ingresar dos valores enteros utilizando una función e imprimirlos por
pantalla
*/
#include <iostream>
using namespace std;

void ingreso(int iNumero1, int iNumero2){
    cout<<"Ingrese el primer valor: ";
    cin>>iNumero1;
    cout<<"Ingrese el segundo valor: ";
    cin>>iNumero2;
}

void main(){
    int iNum1, iNum2;

    system("cls");
    iNum1=0;
    iNum2=5;
    ingreso(iNum1,iNum2);
    cout<<"Primer número ingresado:"<<iNum1<<endl;
    cout<<"Segundo número ingresado:"<<iNum2<<endl;
    fflush(stdin);
    getchar();
}
```

¿Qué sucede con la salida? Porque no se muestran los números ingresados por teclado correctamente?

Buscar la solución, corregir el código y ejecutarlo nuevamente.

Transcribir el siguiente programa, siguiendo las instrucciones planteadas en esta unidad, completando el comentario sobre qué realiza el mismo.
Compilarlo, ejecutarlo y verificar su funcionamiento.

```
/*
Grupo 1
Apellido Nombre, Apellido Nombre,

Ejercicio 0
...
*/
#include <iostream>
#define TOPE 5 // obligatorio para tamaño de arreglos
               // obligatorio en mayusculas
using namespace std;
void cargoVec(int vec[TOPE]){
    for(int i=0;i<TOPE;i++){
        cout<<"Ingrese vec[ "<<i<<" ]=";
        cin>>vec[i];
    }
}

void nuestroVec(int vec[TOPE]){
    for(int i=0;i<TOPE;i++)
        cout<<"vec[ "<<i<<" ]="<< vec[i]<<endl;
}

void main(){
    int vec[TOPE]; // declaracion de un array
    int vec2[TOPE]={10,20,30,40,50}; // definicion de un array
    const float pi=3.14;
    cargoVec(vec);
    nuestroVec(vec);
    fflush(stdin);
    cin.get();
}
```