GUÍA DE USO DE DEV C++

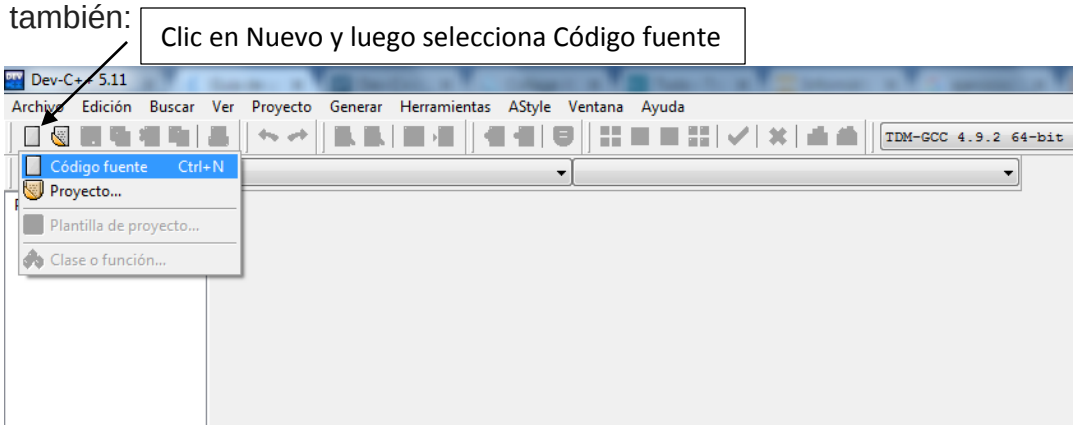
Dev-C++ es software libre y se puede descargar en la siguiente dirección: http://sourceforge.net/projects/orwelldevcpp/

En esta guía básica se explica cómo utilizar ***Dev-C++ 5.11 TDM-GCC 4.9.2*** en *Windows* para, una vez instalado en la computadora, editar, compilar y ejecutar un programa escrito en lenguaje C:

Paso 1: Para crear archivo nuevo

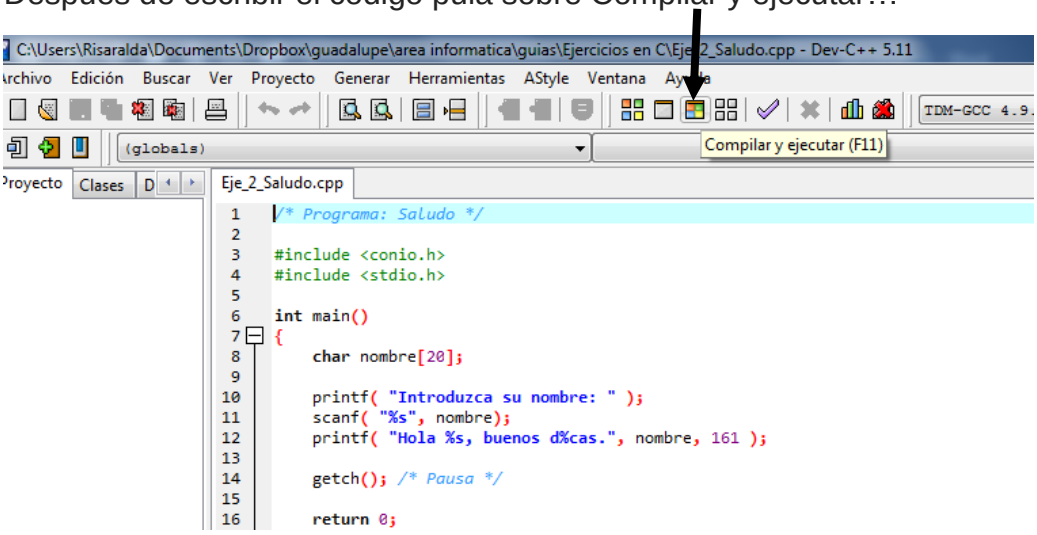
Para iniciar un programa nuevo vaya a: Archivo > Nuevo > Código fuente

O también:



Paso 2: Compilar y Ejecutar

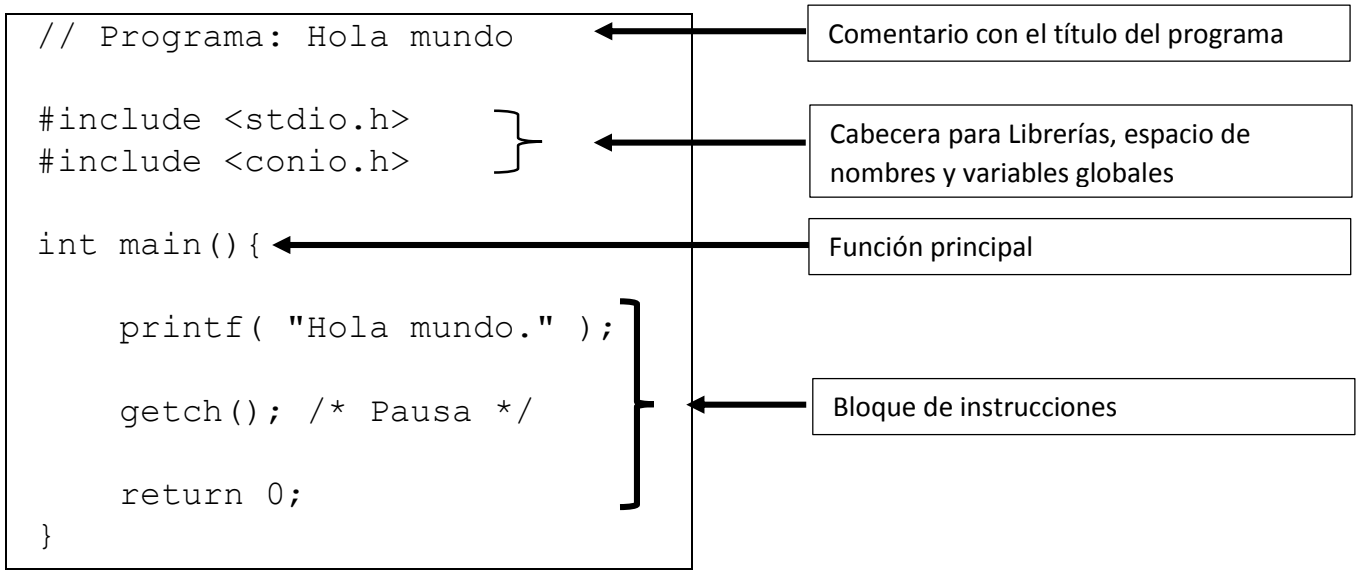
Después de escribir el código pula sobre Compilar y ejecutar…



La primera vez que compile y ejecute, el programa solicita guardarlo. Pone un nombre y selecciona la carpeta donde lo va a guardar.

MI PRIMER PROGRAMA

EJEMPLO 1: Mostrar en pantalla el mensaje Hola Mundo



Consideraciones:

- Comentarios: Son aclaraciones o explicaciones de algo y permiten documentar un código. Los comentarios no se ejecutan en el programa y su uso depende del programador.
    - Para un comentario de una línea se utiliza doble barra inclinada (**//**) al inicio de la línea.
    - Para un comentario de más de una línea, el texto a comentar se ubica entre los caracteres *barra-asterisco* (**/***) y *asterisco-barra* (***/**)
- En la cabecera del programa es necesario incluir librerías o directorios que contienen archivos necesarios para determinadas funciones o declaraciones. Para poder usarlas se usa el comando #include
    - **<stdio.h>** En dicho archivo están escritas las declaraciones de todas las funciones de entrada y salida estándar de la biblioteca estándar de C, como **`printf`** y **`scanf`**
    - **<conio.h>** Para poder hacer uso de **`getch`**
    - **<iostream>** es utilizado para operaciones de entrada/salida. Su nombre es un acrónimo de Input/Output Stream. Incluye funciones como **`cin`** y **`cout`**
- En la cabecera también se puede incluir un espacio para nombres. Para ser más específicos, las palabras reservadas cout y cin están en el

namespace std (standard). Es decir, si deseamos utilizar **cin** y **cout** en un código es conveniente utilizar: **using namespace std;**

- Un programa escrito en C está compuesto por una o más funciones. Existe una función que está presente en todos los programas escritos en C, su misión es marcar el inicio y fin de la ejecución de cada uno de ellos; es la función principal, la primera que se ejecuta; es la función **`main`** y contiene al bloque de instrucciones principal del programa. Su sintaxis "básica" es:

```
int main(){

    <bloque_de_instrucciones>
}
```

- Para mostrar un mensaje en pantalla podemos utilizar dos alternativas:

  - Con la función **printf:**
    - **printf ("mensaje");**

  - Con la función **cout**:
    - **cout<< "mensaje";**

  Nota: Los textos a mostrar van entre comillas y se deben incluir las librerías correspondientes para su funcionamiento.

- La función **`getch`** lee un carácter por teclado, sin eco por pantalla, es decir que cuando se esté ejecutando el programa el carácter oprimido por teclado no se muestra en la pantalla de ejecución. Por tanto, permite realizar una "pausa" antes de cerrar la pantalla de ejecución. Para poder hacer uso de **`getch`** hay que incluir el archivo de cabecera **`conio.h`**.

- Con **`return 0`** se está informando al sistema operativo, donde se ejecute el programa, que este finalizó correctamente.

- Obsérvese que, en lenguaje C, las instrucciones de expresión siempre van seguidas de un *punto y coma* (`;`), el cual indica que la instrucción ha terminado.

EJEMPLO 1 MODIFICADO

Mostrar en pantalla el mensaje "Hola mundo." una línea más abajo, dejando 3 espacios en blanco al principio de la línea y, además, se desea mostrar el mensaje "Pulse una tecla para salir...":

```
/* Programa: Hola mundo (Versión 2) */

#include <conio.h>
#include <stdio.h>

int main(){

    printf( "\n   Hola mundo." );
    printf( "\n\n   Pulse una tecla para salir..." );

    getch(); /* Pausa */

    return 0;
}
```

Consideraciones:

- **\n** dentro de **printf** sirve para mover el cursor al principio de la línea siguiente
- Con **cout** la instrucción sería **<<endl**

EJEMPLO 2: Escribir en lenguaje C un programa que:

1) Pida por teclado el nombre de una persona.

2) Muestre por pantalla el mensaje: "Hola <nombre>, buenos días.".

SOLUCIÓN:

```
// Programa: Saludo

#include <conio.h>
#include <stdio.h>

int main()
{
    char nombre[20];

    printf( "Introduzca su nombre: " );
    scanf( "%s", nombre );
    printf( "Hola %s, buenos dias.", nombre);

    getch(); /* Pausa */

    return 0;
}
```

Consideraciones:

- **char nombre[20]** sirve para declarar la variable **nombre** de tipo cadena, pudiendo contener 20 **char** (caracteres).
- Cuando se quiere almacenar un dato se utiliza el comando scanf. El especificador de formato **%s** debe utilizarse tanto en **scanf** como en **printf** para las cadenas (*strings*).

EJEMPLO 3: Se pide, en lenguaje C, escribir un programa que:

1) Pida por teclado el lado de un cuadrado.
2) Calcule el perímetro del cuadrado.
3) Muestre por pantalla el resultado del perímetro.

SOLUCIÓN

```
/* Programa: Perimetro de un cuadrado (Solución 1) */

#include <conio.h>
#include <stdio.h>

int main()
{
    int lado, area;

    printf("\n Introduzca el lado de un cuadrado: ");
    scanf( "%d", &lado );

    area = lado+lado+lado+lado;

    printf( "\n   El area del cuadrado es: %d",area);

    printf( "\n\n   Pulse una tecla para salir..." );
    getch(); /* Pausa */

    return 0;
}
```

Consideraciones

- **int lado, area;** sirve para declarar las variables **lado** y **area** con el fin de almacenar dos números enteros (**int**).
- **scanf("%d", &lado)** permite que el usuario del programa introduzca por teclado el valor del lado y, puesto que la variable **lado** es de tipo **int** (número entero), se tiene que escribir el especificador de formato asociado a la entrada de un número real (**%d**). Por otra parte, el carácter *ampersand* (**&**) sirve para indicar la dirección de memoria de la variable **lado**, es decir, la dirección de memoria donde se va a almacenar el dato introducido por el usuario.
- Al escribir **area = lado+lado+lado+lado** se asigna a la variable **area** el resultado de aplicar la fórmula del perímetro del cuadrado.

CONSIDERACIONES GENERALES DEL LENGUAJE C

TIPOS DE VARIABLES MÁS HABITUALES

TIPO DE DATOS	SE ESCRIBE	MEMORIA REQUERIDA*	RANGO ORIENTATIVO*	EQUIVALENCIA EN PSEUDOCÓDIGO	OBSERVACIONES
Entero	int	2 bytes	- 32768 a 32767	Entero	Uso en contadores, control de bucles etc.
Entero largo	long	4 bytes	- 2147483648 a 2147483647	Entero	Igual que int pero admite un rango más amplio
Decimal simple	float	4 bytes	$-3{,}4\cdot10^{38}$ a $3{,}4\cdot10^{38}$	Real	Hasta 6 decimales. También admite enteros
Decimal doble	double	8 bytes	$-1{,}79\cdot10^{308}$ a $1{,}79\cdot10^{308}$	Real	Hasta 14 decimales. También admite enteros
Carácter	char	1 bytes	0 a 255	Alfanumérica	Carácter, independiente o parte de una cadena

* Podría variar

Especificador de formatos habilitados para trabajar con printf y scanf:

Especificador	Descripción
%c	Carácter
%d	Número entero(int)
%i	Número entero(int)
%D	Número entero long(o también %ld)
%f	Punto flotante(float)
%e	Notación científica con e minúscula
%E	Notación científica con E mayúscula
%g	Formato para tipo punto flotante(float)
%G	Formato para tipo punto flotante(float)
%o	Número octal sin signo
%s	Cadena de texto
%u	Entero sin signo
%U	Entero sin signo long(o también %lu)
%x	Hexadecimal sin signo con minúsculas
%X	Hexadecimal sin signo con mayúsculas
%p	Puntero, dirección de memoria
%n	Número de caracteres
%o	Formato entero octal
%O	Formato entero octal long(o también %lo)
%lf	Formato double
%LF	Formato long double
%l	Formato double
%h	Formato double
%L	Formato long double

ACTIVIDADES

EJERCICIO 1: Escribir un código en C que muestre el mensaje: Hola Mundo

EJERCICIO 2: Escribir en lenguaje C un programa que:

1) Pida por teclado el nombre de una persona.

2) Muestre por pantalla el mensaje: "Hola <nombre>, bienvenido.".

EJERCICIO 3: Se pide, en lenguaje C, escribir un programa que:

1) Pida por teclado el lado de un cuadrado.
2) Calcule el perímetro del cuadrado.
3) Muestre por pantalla el resultado del perímetro.

EJERCICIO 4: Hallar el perímetro y el área de un rectángulo si se pide la longitud del lado más largo y la longitud del lado más corto

EJERCICIO 5: Conocido el diámetro de una circunferencia mostrar su área y su perímetro (datos de tipo float).

EJERCICIO 6: Escribir en lenguaje C un programa que:

1) Pida por teclado dos números (datos enteros).

2) Calcule la suma y multiplicación de los dos números introducidos.

3) Muestre por pantalla los resultados (datos enteros).

EJERCICIO 7: Escribir en lenguaje C un programa que:

1) Pida por teclado la nota de tres exámenes (datos float).

2) Calcule la nota promedio de los tres exámenes.

3) Muestre por pantalla el resultado (dato float).

EJERCICIO 8: Escribir un programa en C que ingresada una cantidad de dólares americanos me muestre su equivalente en pesos colombianos.

EJERCICIO 9: Escribir un programa en C que ingresada una cantidad de Pesos colombianos me muestre su equivalente en Dólares americanos.

EJERCICIO 10: Escribir en lenguaje C un programa que:

1) Pida por teclado dos números (datos enteros) y sean almacenados en dos variables, llamadas *v1* y *v2*.

2) Intercambie los valores de las variables.

3) Muestre por pantalla los valores contendidos en las variables.

En pantalla se vería así: