

GRÁFICOS EN DEV C++

GUÍA

PAOLA CORTEZ

PASOS INICIALES

En el siguiente link (<http://usuarios.multimania.es/charlytospage/dev.htm>)descargar la librería Winbgim

Links Relacionados.

[Home page del programador de winbgim](#)



[Descargar instalador Winbgim for Dev C++](#)

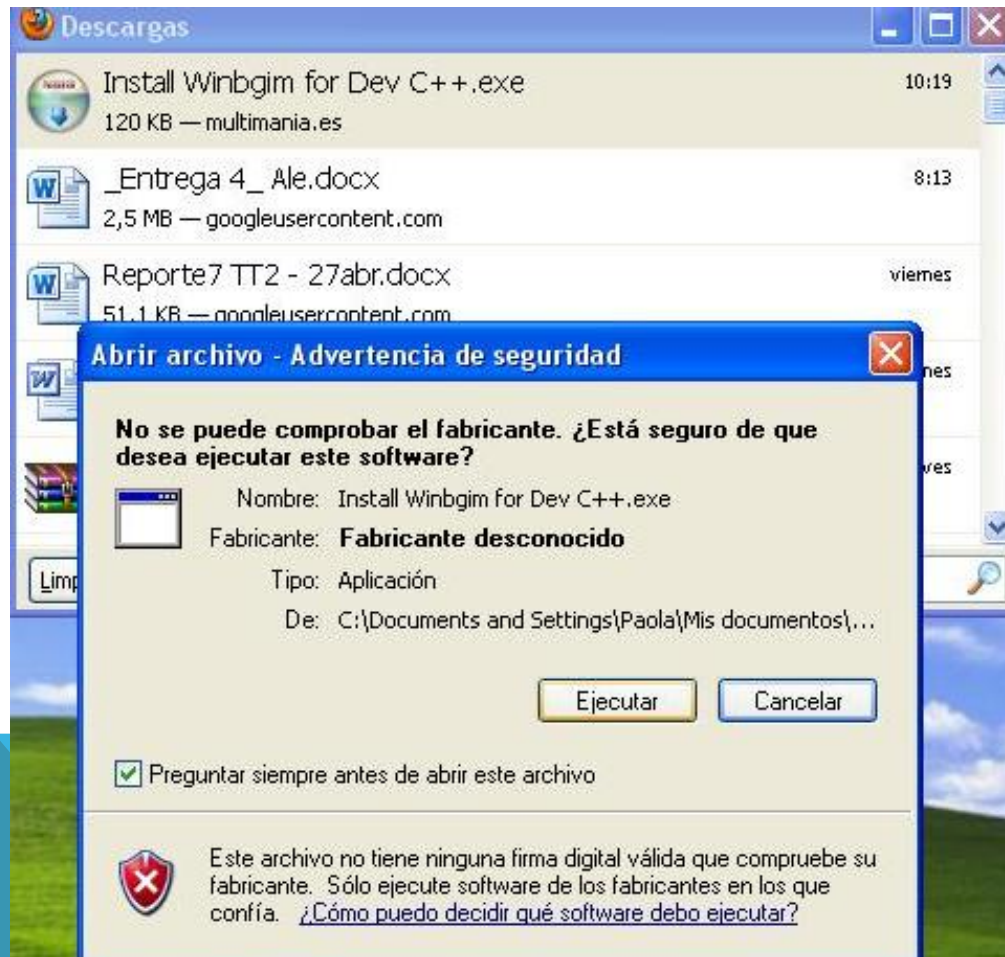


[Descargar copia del manual en pdf](#)

Instalarla, observe que aparece un asistente en donde debe de aceptar iniciar la instalación.

CONTINUACIÓN..

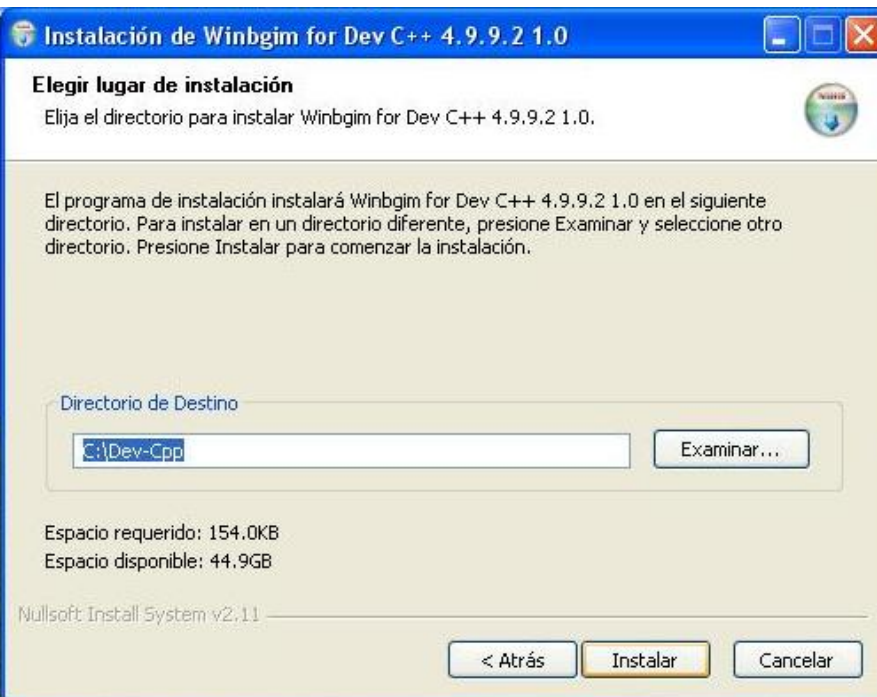
Inicie el proceso de instalación



CONTINUACIÓN...



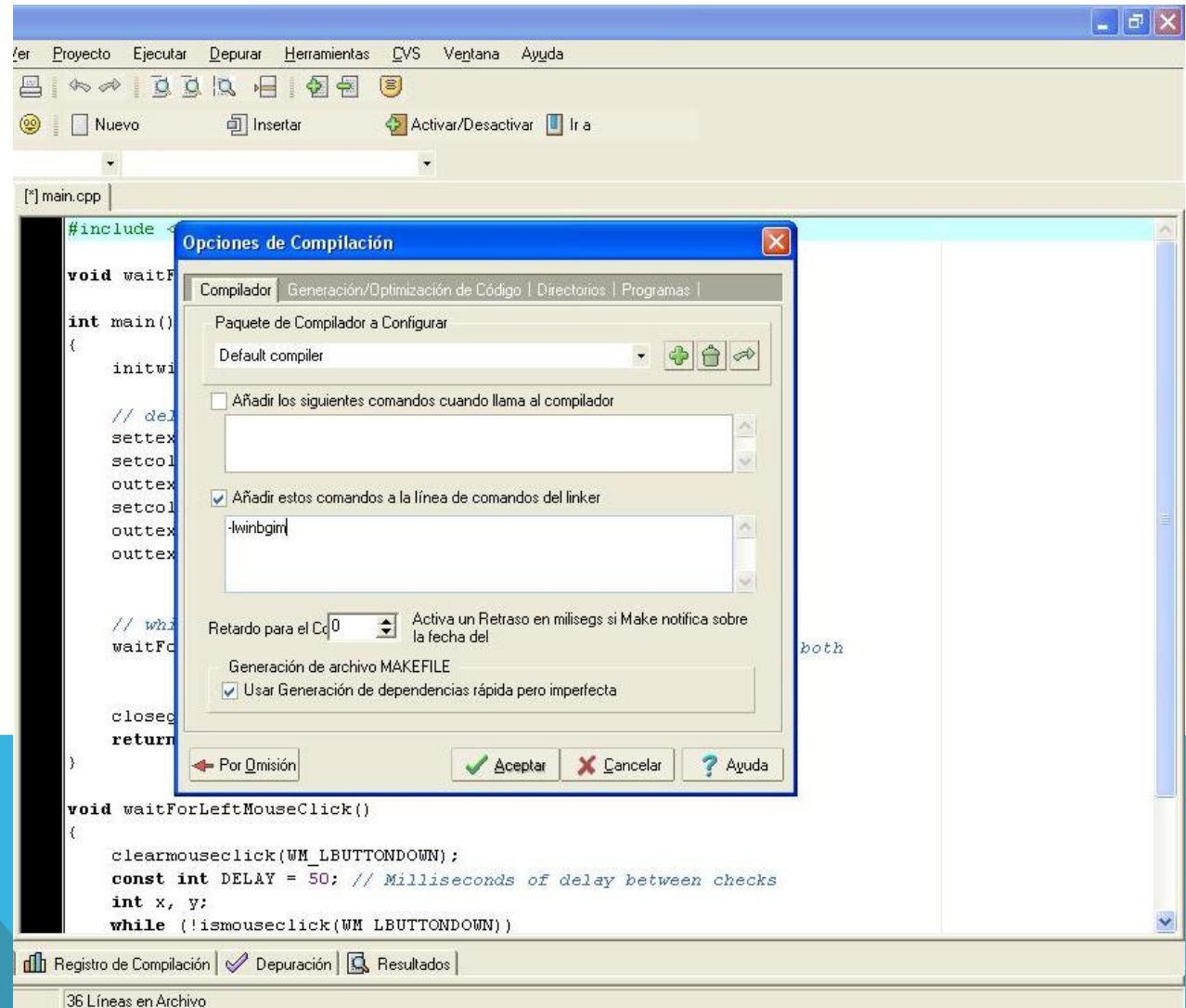
CONTINUACIÓN...



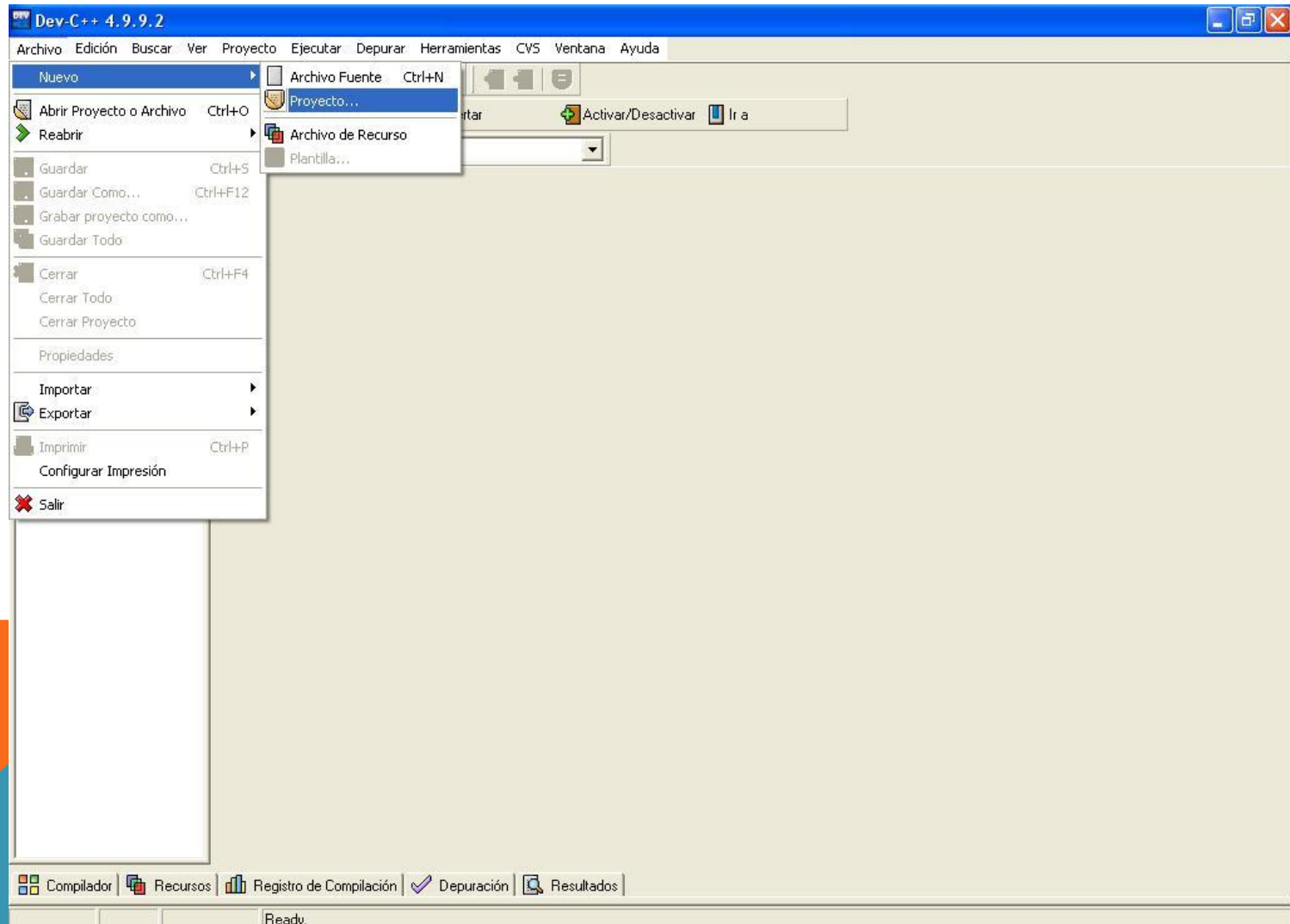
CONTINUACIÓN

Una vez instalada dentro del compilador, se debe dirigir al menú Herramientas,

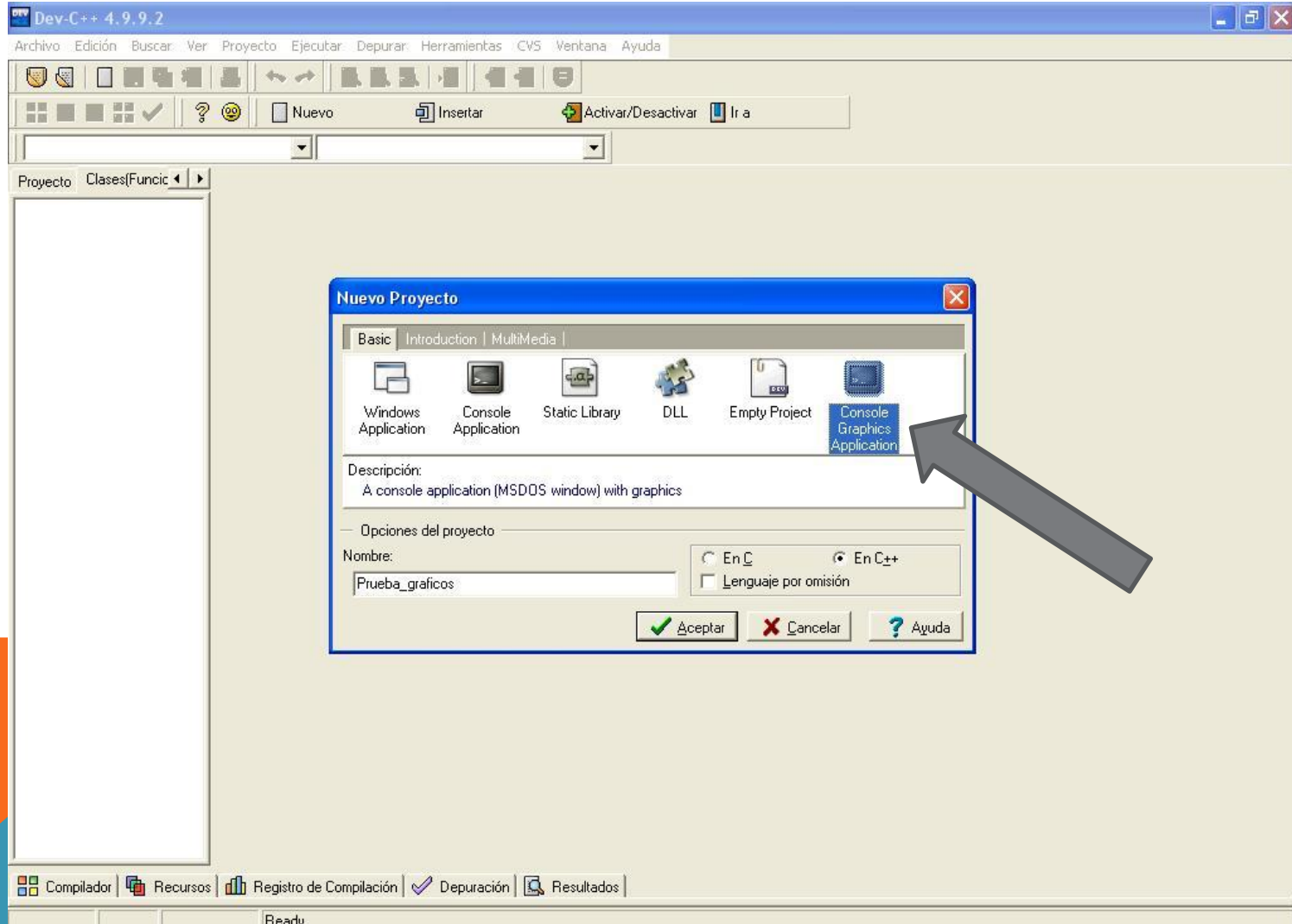
Opciones del compilador y ahí escriba (el nombre de la librería) como se muestra a continuación:



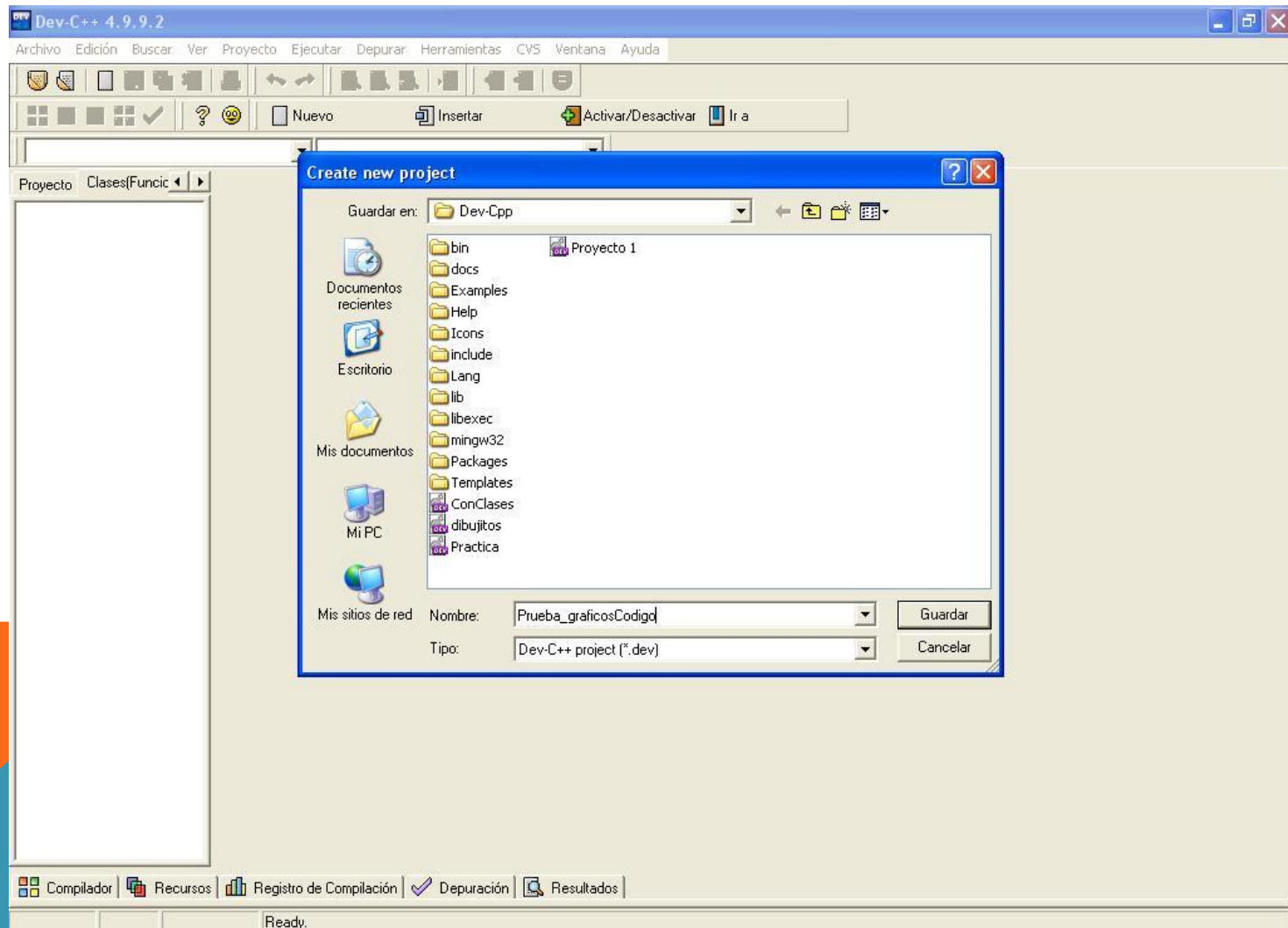
CREAR UN NUEVO PROYECTO



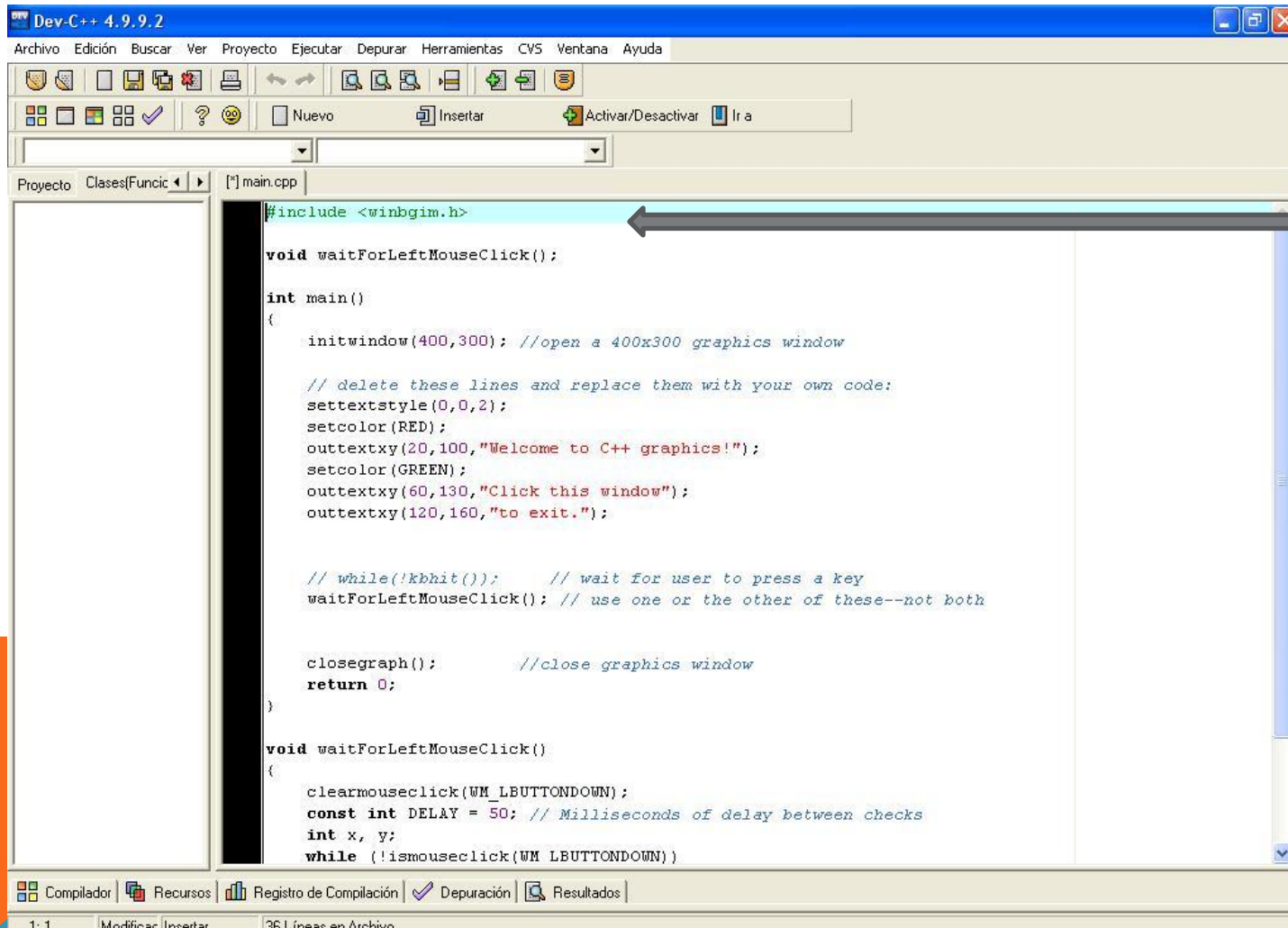
SELECCIONAR LA CONSOLA GRAFICA



GUARDAR EL PROYECTO



CÓDIGO GENERADO POR DEFAULT



```
#include <winbgim.h>

void waitForLeftMouseClicked();

int main()
{
    initwindow(400,300); //open a 400x300 graphics window

    // delete these lines and replace them with your own code:
    settextstyle(0,0,2);
    setcolor(RED);
    outtextxy(20,100,"Welcome to C++ graphics!");
    setcolor(GREEN);
    outtextxy(60,130,"Click this window");
    outtextxy(120,160,"to exit.");

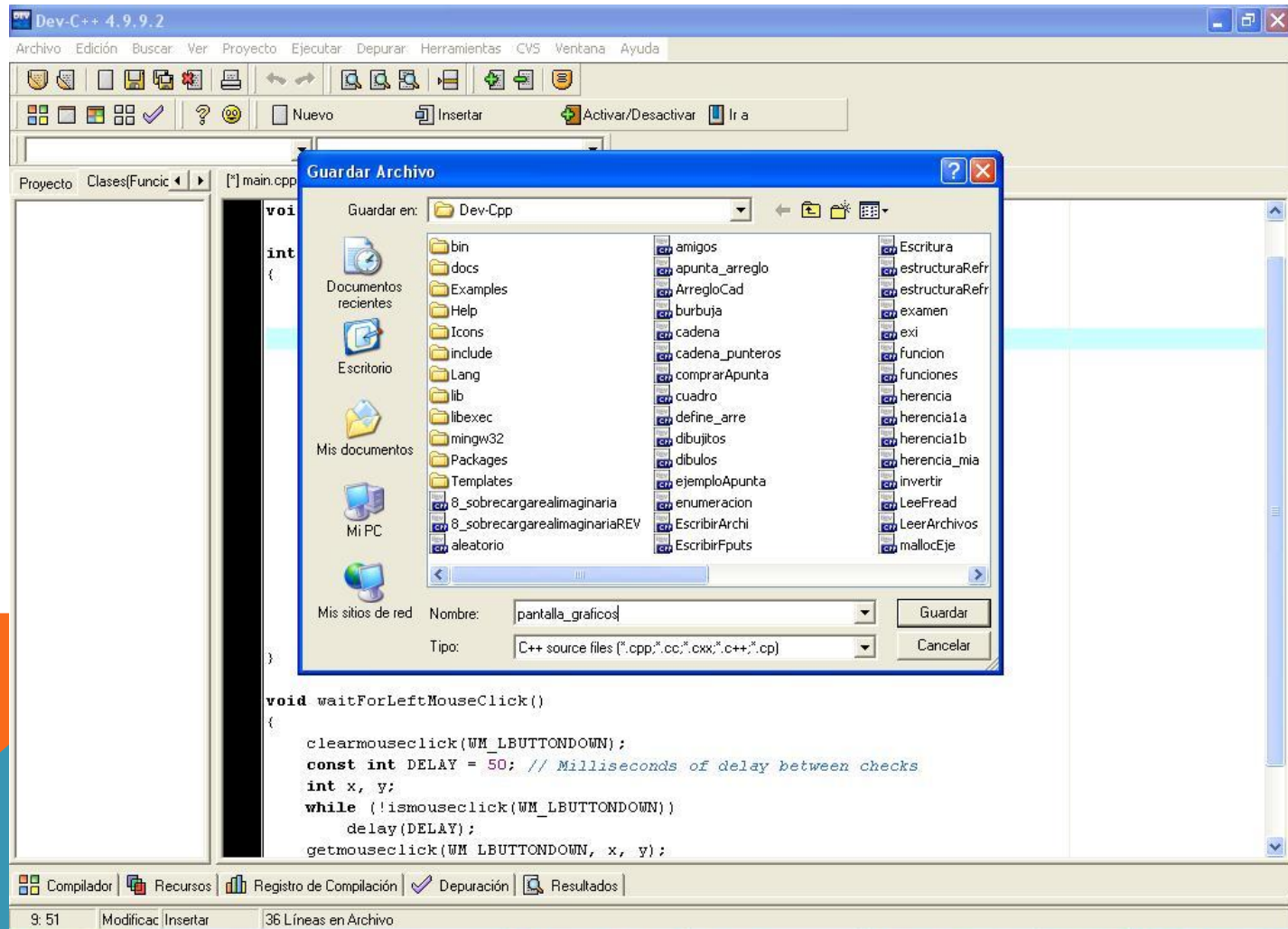
    // while(!kbhit()); // wait for user to press a key
    waitForLeftMouseClicked(); // use one or the other of these--not both

    closegraph(); //close graphics window
    return 0;
}

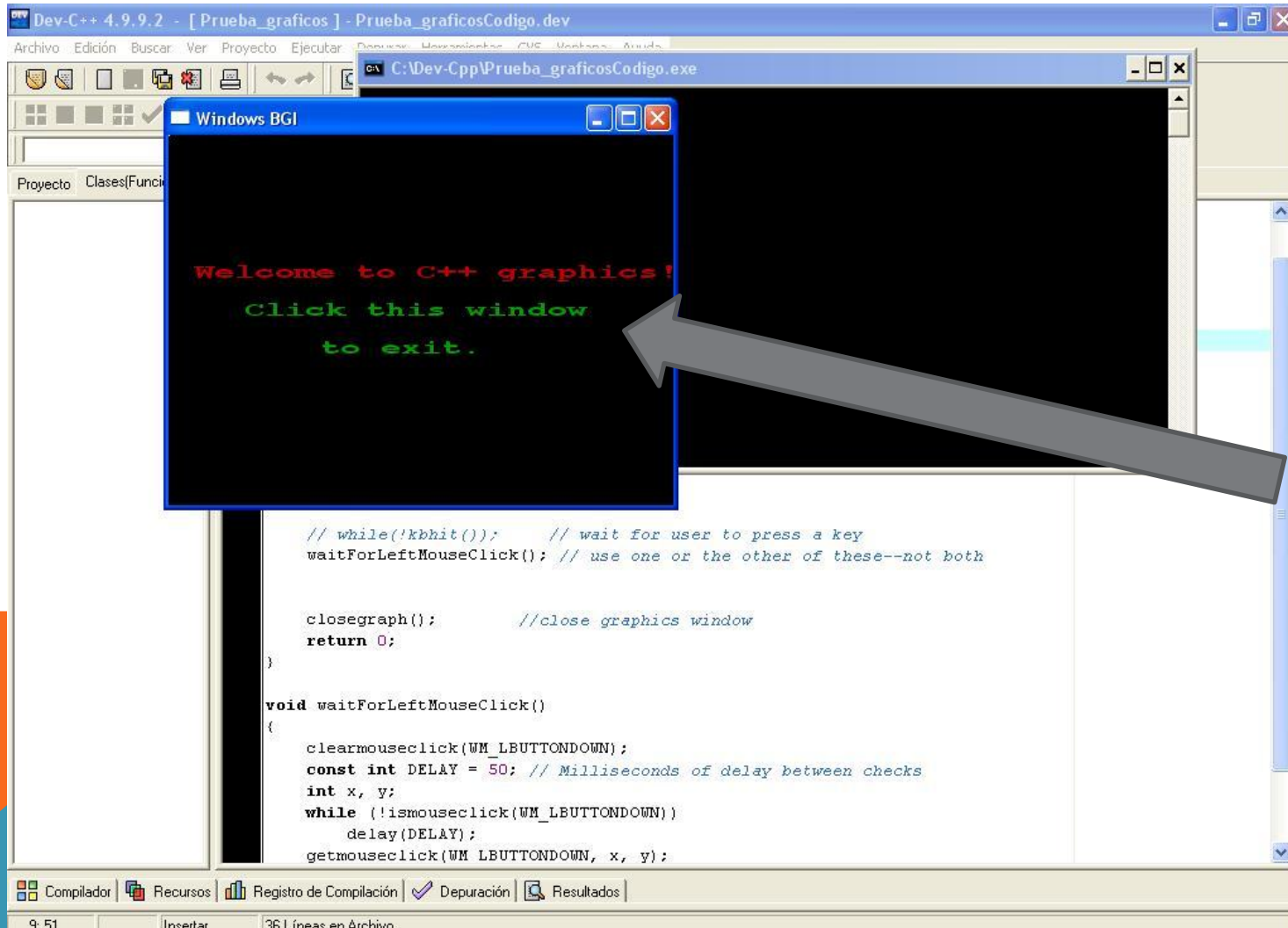
void waitForLeftMouseClicked()
{
    clearmouseclick(WM_LBUTTONDOWN);
    const int DELAY = 50; // Milliseconds of delay between checks
    int x, y;
    while (!ismouseclick(WM_LBUTTONDOWN))
```

Note el uso de la librería `<winbgim.h>`

AL EJECUTAR, GUARDAR EL CÓDIGO



EJECUCIÓN



Pantalla con los gráficos

VERSIÓN DEL DEV C++



Para instalar la librería de gráficos (<winbgim.h>) en Dev C++, recomiendo este sitio:

<http://usuarios.multimania.es/charlytospage/dev.htm>

OTRAS FUNCIONES PARA DIBUJAR

`line (x1,y1,x2,y2);`

`rectangle (x1,y1,x2,y2);`

`bar (x1,y1,x2,y2);`

`circle (x,y,radio);`

** En Turbo C, los pixeles de la pantalla van sobre el eje 'x' 640, el eje 'y' 480*

**En Dev C++, con la función `initwindow(x,y)` el programador indica la cantidad de pixeles en los ejes.*

