



Instalación del Dev C++ 4.9.9.2.

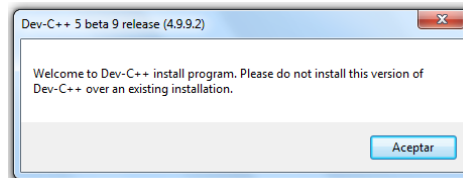
Alumno: Mario Gutierrez Leal
Número de cuenta: 411078470

Bloque: 150

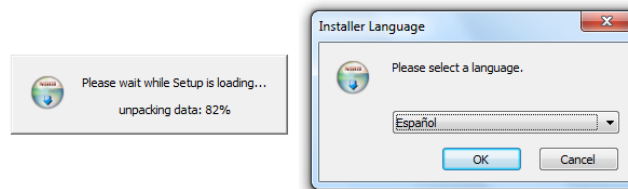
En primera instancia, descargamos la paquetería del portal del distribuidor.

<http://www.bloodshed.net/dev>

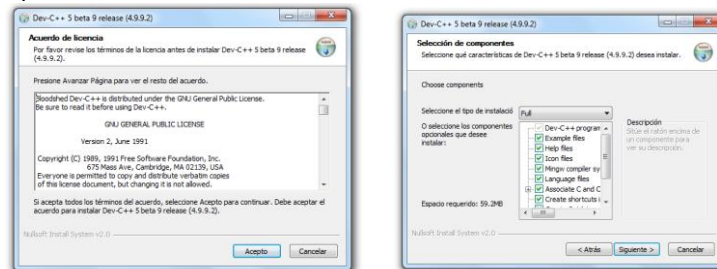
Una vez descargado el archivo “*devcpp-4.9.9.2_setup.exe*” lo ejecutamos normalmente (dando los permisos necesarios para que modifique en menor medida la configuración).



Dejamos que se descomprima la información necesaria del setup y elegimos el idioma que deseemos.

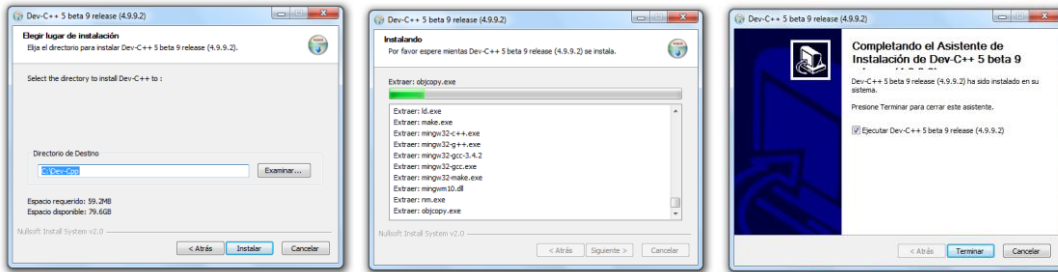


Aceptamos los términos legales de la instalación del dev c++ 5 beta. Para posteriormente seleccionar los componentes a instalar. Como recomendación, de preferencia elegir “full” (en el caso del Dev C++ “costum” ya que existe una variante para la creación de GUI, muy al estilo del Borland C++ Builder).



Elegimos el directorio en donde se instalara nuestra versión del DevC. Normalmente es la unidad C:/; claro que esto varía en cuestión del número de discos duros y particiones con las que se cuenta el equipo.

Indicamos que queremos continuar con la instalación y esperamos a que termine la extracción de los archivos, el directorio ya indicado.



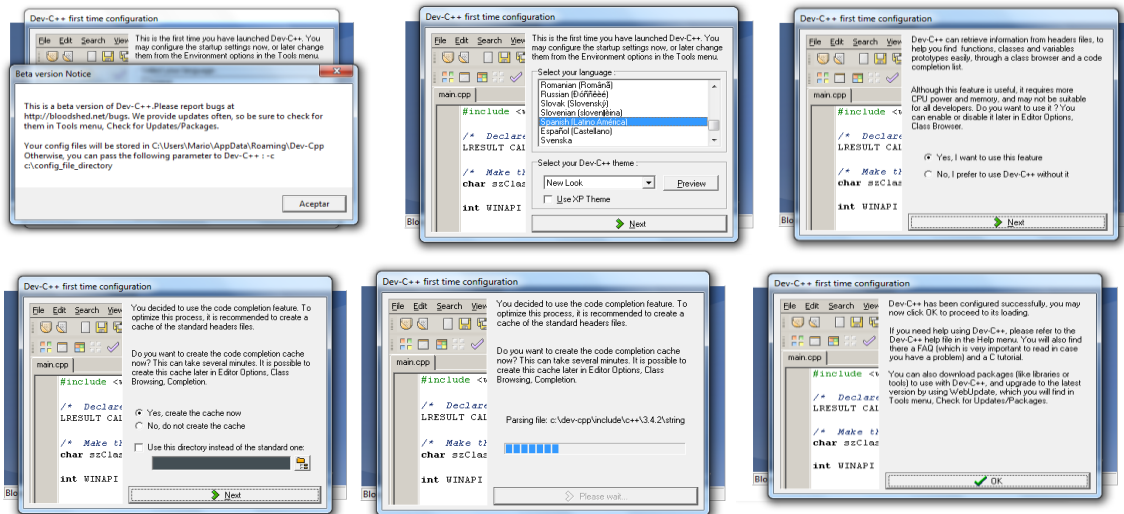
Ahora podemos hacer uso del Dev C++, con sus archivos cabeceras nativas. Sin embargo, resulta conveniente realizar la instalación de algunos archivos cabeceras muy útiles de Borland (Cuidado: Estos archivos cabecera tiene derechos de autor).

Para lo cual descargamos *conio.h*, de internet (en mi caso ya la tengo instalada). Estos archivos los instalamos en:



Una vez instalada la cabecera *conio.h*, ya estamos listos para la configuración del Dev C++.

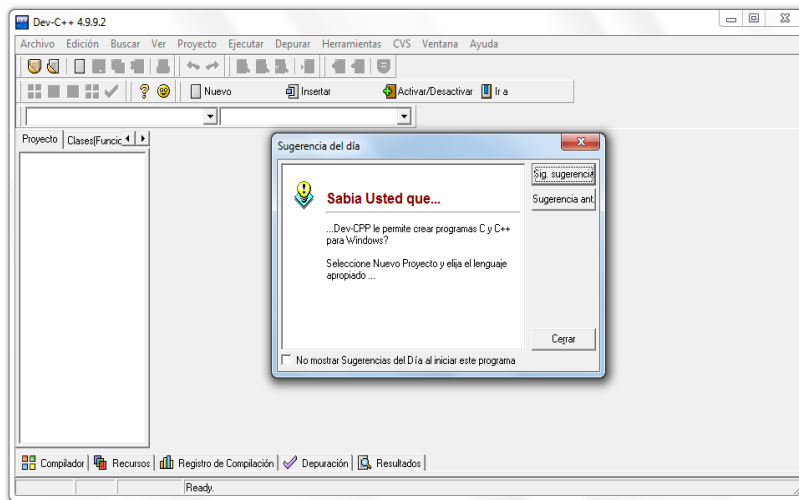
Al realizar la configuración inicial, se nos indica que la versión aun no es la definitiva y por lo cual, requieren arreglar algunos bugs (en realidad esta versión ya cuenta con más de 2 años publicada, sin cambio aparente). También se hace un parseo de las cabeceras, para no redundar en ellas y también se confirma el idioma de preferencia.



Ahora estamos listos para hacer uso del DevC++ y su magnífica interfaz grafica.

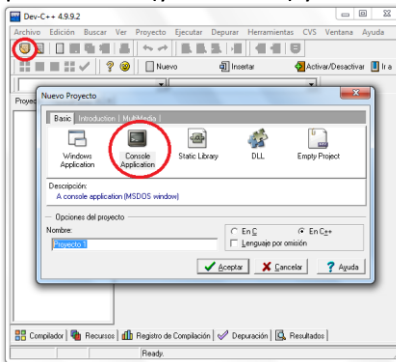


Esta es la interfaz grafica del DevC++ 5 Beta ya que en versiones anteriores se usa la interfaz gnome (nomo). En cuestión de la sección “Sabia usted que...”, la prefiero desactivar, a la larga comienza a molestar más que ayudar y así ahorramos un poco de tiempo al iniciar o continuar nuestro proyecto.

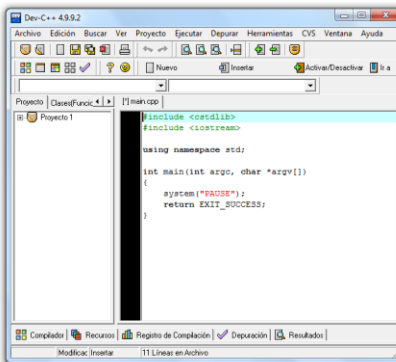


Para crear un nuevo programa, debemos crear un nuevo proyecto. En comparación con otros compiladores como TurboC (de Borland me parece) o Eclipse (para Linux), DevC++ engloba varios archivos de C/C++ en el mismo proyecto, lo que nos ayuda a tener una buena gestión en proyectos y más en el desarrollo de aplicaciones, así como para creación del correspondiente diagrama UML.

Para comenzar, elegimos las opciones de nuevo proyecto y la creación de una aplicación de consola, así como el lenguaje de preferencia (yo uso C++) y finalmente el nombre del proyecto.



Una vez creado el proyecto 1, tenemos lo siguiente en el DevC++:



Un vistazo más cercano:

```
#include <cstdlib> //Archivos cabecera:
#include <iostream> // Son necesarios para compilar le programa y le decimos que ocupar.

using namespace std; // Indica que donde hay un "gap" se pueda usar Std, así no redundamos con std::cout ni std::cin.

int main(int argc, char *argv[]) // El programa principal "main" el cual regresa un valor tipo entero.
{
    system("PAUSE"); //Pausa antes de terminar el proceso, el cual pasa hasta que se registre un char*.
    return EXIT_SUCCESS; //retorno y salida del "main".
}
```

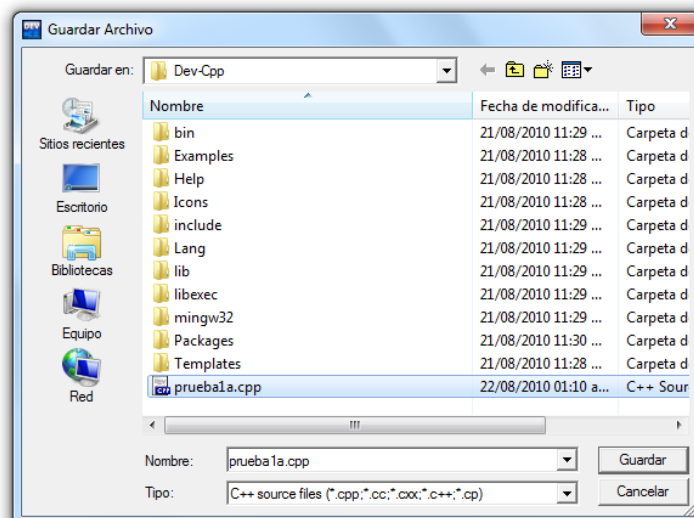
Modificamos un poco el programa para poder mandar un mensaje:


```
#include "conio.h" //Conio.h no es nativa del Devc++ por eso está en comillas y es necesaria para usar getch().
#include <iostream> //Maneja la entrada y salida de datos, en este caso el despliegue con cout;

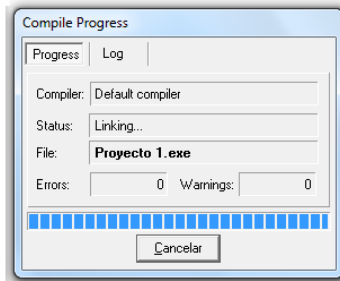
using namespace std;

int main()
{
    cout << "\t\t Universidad Nacional Autónoma de México\n" //Imprimir el texto en pantalla.
         "\t\t Facultad de Ingeniería\n" // "\t" significa tabular el texto
         "\t\t By Gutierrez Leal Mario\n" // "\n" indica una nueva línea.
         ""
         " \n\t\t Teclea para salir... ";
    getch(); //Espera el ingreso de un valor, para continuar al siguiente proceso. En este caso es el fin del programa.
    return 0;
}
```

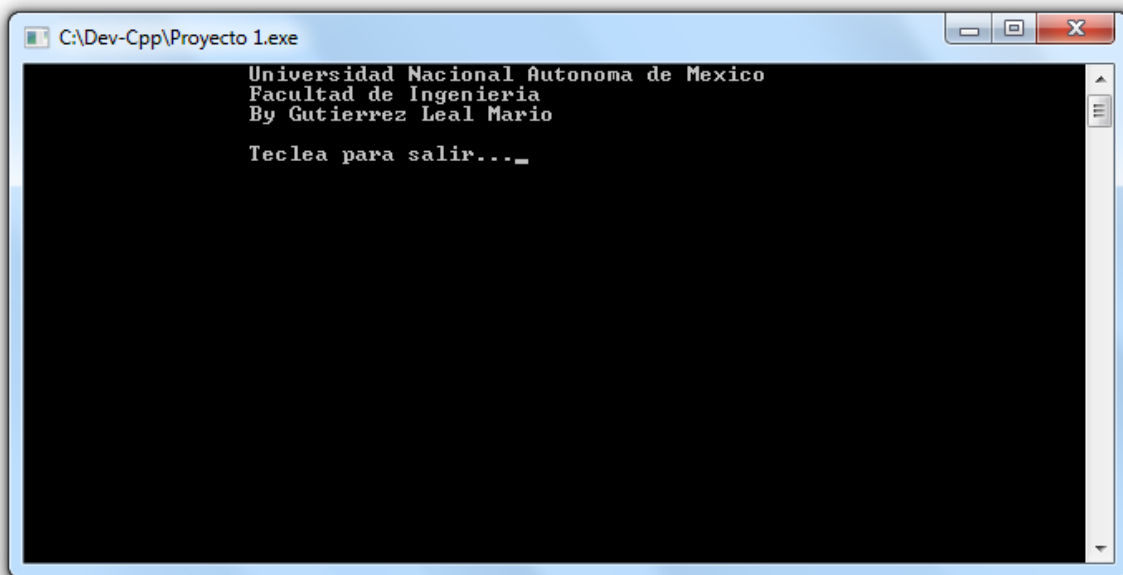
Guardamos el archivo como "prueba1a.cpp". Hay que tener cuidado con no guardarlo como "main.cpp" (así lo tiene de default) ya que este error es muy común en los novatos y crea un error redundancia con el programa principal "main" y el archivo "main.cpp".



Finalmente lo compilamos y ejecutamos con el botón  :



En este ejemplo sencillo, no tenemos errores así que no hacemos uso del debug para evidenciarlos y corregirlos. Por lo tanto tenemos como resultado nuestro programa.



Bibliografía.

- Walter Savitch. *Absolute C++ 2th*. Addison Wesley USA. Pág 14 y 15.