Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

Facultad de Ingeniería y Ciencias Hídricas
FICH
UNL

FUNDAMENTOS DE PROGRAMACIÓN

Asignatura correspondiente al plan de estudios de la carrera de Ingeniería Informática

ANEXO 1
Introducción a las Herramientas de Desarrollo

Ing. Pablo Novara - 19/04/2010

Introducción a las Herramientas de Desarrollo

Herramientas a utilizar

Para desarrollar un programa utilizando C++ sólo se requiere un editor de textos para poder escribir el código fuente (por ejemplo el Block de Notas de Windows) y un compilador (en el sentido amplio de la palabra, suele conformarse por varios programas) para poder generar el ejecutable. Es decir, el compilador toma un archivo de texto con un código fuente, verifica si la sintaxis es correcta y realiza los pasos necesarios para generar el archivo ejecutable correspondiente a dicho código (traducir de C++ a lenguaje de máquina). Sin embargo, en la práctica se suele utilizar alguna herramienta que integre un editor y un compilador, junto con muchas características adicionales destinadas a brindar comodidad y velocidad al programador. Estas herramientas se denominan IDEs (del inglés: Integrated Development Enviroment = Entorno Integrado de Desarrollo).

Un IDE incluye una interfaz visual, que permite trabajar con comodidad y que se encarga de dialogar internamente con el compilador, el enlazador y demás herramientas. De esta forma, el programador nunca debe llamarlas directamente, sino que con solo hacer un click, la IDE genera y ejecuta todos los comandos necesarios para la compilación y presenta los resultados de forma adecuada. Esto, además de ser más rápido, evita al programador la necesidad de recordar los numerosos parámetros que el compilador y el enlazador reciben y los reemplaza por un cuadro de diálogo mucho más simple. De igual manera, simplifica otras tareas como la depuración (ejecución paso a paso, inspección de variables, etc.), o la edición, y suele presentar además, diferentes asistencias para la escritura del programa, como sugerencias de autocompletado, coloreado de la sintaxis del código fuente, ayuda acerca del lenguaje, etc.

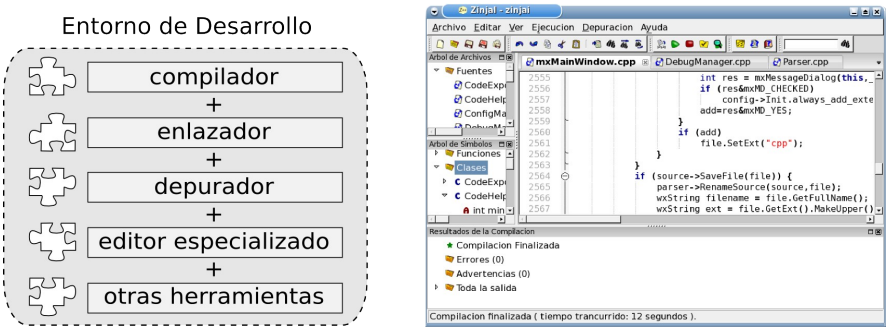


Figura 1: Izquierda: estructura de un IDE. Derecha: ejemplo de IDE (Zinjal)

Es común para el usuario inexperto confundir un IDE con un compilador, ya que en la mayoría de los productos de desarrollo todo el conjunto de herramientas necesarias se proveen e instalan junto con la IDE, y puede que el programador jamás necesite entrar en contacto directo con estas.

Ventajas y desventajas del uso de una IDE

Cuando el alumno realiza su primeros pasos en un lenguaje de programación es altamente recomendable que utilice una IDE. De no hacerlo, deberá lidiar con cuestiones relacionadas al proceso de compilación y al manejo de la linea de comandos para el uso de un compilador en particular. Es conveniente, en una primer etapa, concentrarse en las cuestiones relacionadas al lenguaje y a la lógica de los problemas a resolver.

Sin embargo, un programador avanzado debe conocer con cierto detalle qué ocurre detrás de su IDE, y cómo se gestiona internamente el proceso de compilación. Estos conocimientos le ayudan a entender con mayor grado de profundidad el lenguaje, y le permiten abordar proyectos de mayor envergadura donde intervengan distintos lenguajes, bibliotecas externas, se requiera soporte para múltiples plataformas, etc.

En esta primer guía relacionada a las herramientas para la programación en C++, nos centraremos sólo en la primer parte (el uso de una IDE). El conocimiento adicional sobre el proceso de compilación será tema de otro anexo más adelante cuando el alumno disponga de cierto grado de experiencia con el lenguaje.

En este curso, cada alumno es libre de utilizar el IDE que desee. Sin embargo, en esta guía se presentará uno de ellos (Zinjal), que tiene la ventaja de estar desarrollado dentro de la cátedra y pensando para su uso en el aula por parte de estudiantes, además de ser de libre distribución.

Introducción al uso de Zinjal

Zinjal es un IDE para programar en C++ inicialmente desarrollado para el dictado de clases y para ser utilizado por estudiantes. Invoca internamente al compilador GCC para generar los ejecutables y se encuentra disponible, como Software Libre, tanto para Windows como para GNU/Linux. Cuenta con numerosas facilidades de edición y asistencias para la codificación, así como un sistema de depuración integrado y ayuda en castellano. Se puede descargar desde "http://zinjai.sourceforge.net"[1].

1 Zinjal es un software en constante desarrollo, por lo que es actualizado frecuentemente. Se recomienda que visite el sitio aunque ya disponga de una versión anterior de este IDE para verificar si no existen nuevas actualizaciones.

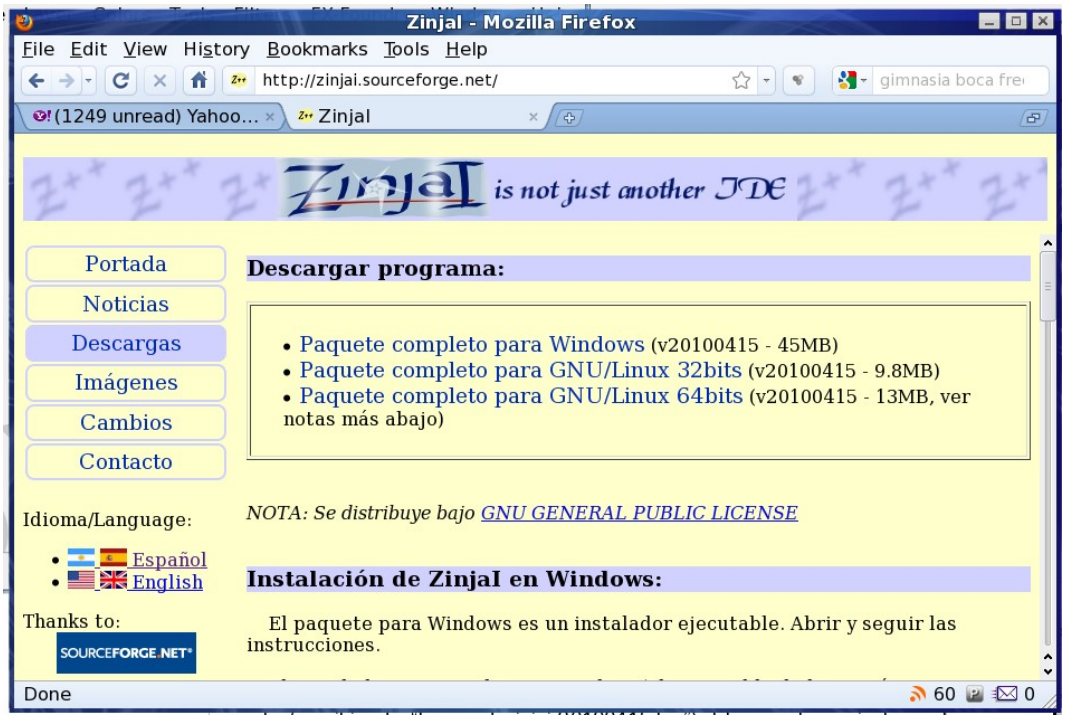


Figura 2: Página de descarga de Zinjal

El proceso de instalación en Windows es similar a la instalación de cualquier otro programa. Para una instalación normal se debe ejecutar el instalador, aceptar la licencia y presionar siguiente tres veces. El instalador copiará los archivos del IDE, así como también de MinGW (compilador, depurador, etc.).

En GNU/Linux, no es necesario un proceso de instalación del IDE. Sólo se debe descomprimir el archivo, y lanzar el ejecutable zinjai dentro de la carpeta con el mismo nombre que se creará al descomprimir. Para descomprimirlo, puede utilizar alguna herramienta gráfica (como ark o file-roller), o desde una consola (escribiendo por ejemplo “tar -xzvf zinjai-l64-20100415.tgz”). El paquete no incluye al compilador (g++) ni al depurador (gdb). Los mismos, deberán ser instalados con el gestor de paquetes que corresponda según la distribución (aptitude, sinaptics, installpkg, yum, etc.).

Zinjal presenta dos modos de trabajo:

- Cuando se inicia Zinjal se encuentra en un modo pensado para desarrollar rápidamente ejercicios simples. Aquí, cada pestaña (cada archivo abierto) será considerado un programa diferente, al compilar y ejecutar, se considera sólo la pestaña actual. Permite trabajar sin necesidad de crear un proyecto ni realizar configuración alguna. Tampoco es necesario (aunque si recomendable) que grabe su código fuente. Cuando se crea un programa simple, se crea un único archivo. El ejecutable de un programa simple, será un archivo con su mismo nombre, pero con extensión .bin o .exe (según se utilice GNU/Linux o Windows) en el mismo directorio (carpeta) que el fuente.
- El otro modo, que se utiliza cuando se abre un proyecto, no permite tener más de un programa abierto al mismo tiempo, y todos los fuentes con que se trabaje pertenecerán a un mismo proyecto. El proceso de compilación tendrá en cuenta todos los archivos que pertenecen al proyecto, y las posibilidades de configuración del mismo serán mucho más amplias que en el caso anterior. Cuando se crea un proyecto, se crea un nuevo directorio, en el cual se guardarán el archivo de

configuración del proyecto, los archivos objetos y el ejecutable, y todos los fuentes, cabeceras y otros archivos que el usuario cree dentro de su proyecto.

A continuación se desarrollará paso a paso un ejemplo para crear un programa simple. Este ejemplo está basado en uno de los tutoriales de Zinjal. Puede encontrar más y conocer otras características accediendo a los mismos a través del menú de ayuda del programa.

Primer programa con Zinjal

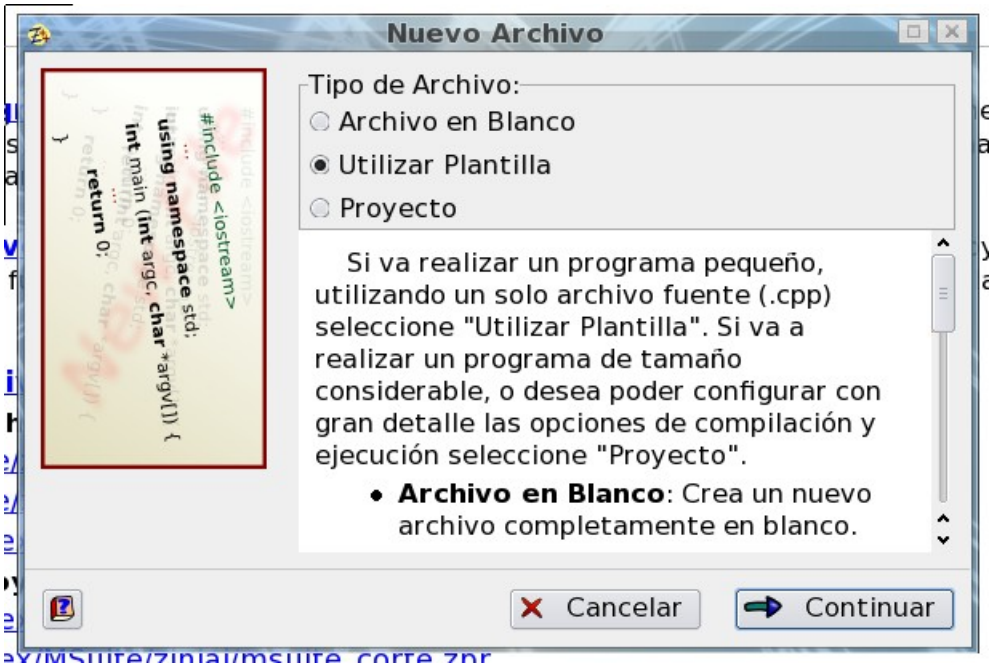
Vamos a resolver un pequeño ejercicio para ejemplificar el manejo básico de la interfaz. El enunciado del ejercicio sería:

> *Realice un programa que permita calcular las raíces de una función cuadrática.*

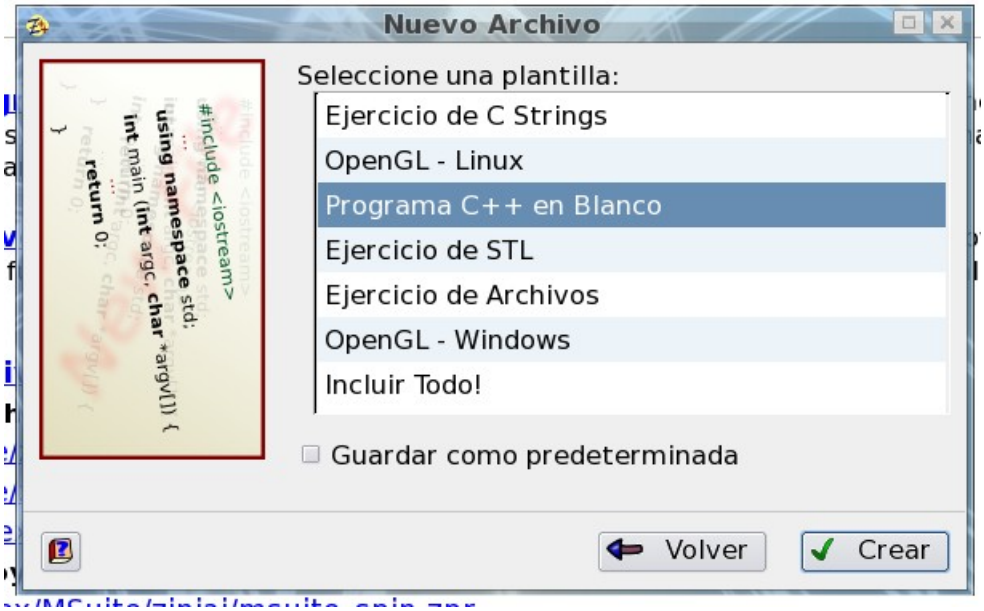
Paso 1:

Lo primero que debe hacer, es crear un nuevo programa. Para ello seleccione la opción *Nuevo...* del menú *Archivo*.

Se desplegará inmediatamente el Asistente para Nuevo Archivo. Allí seleccione la opción *Utilizar Plantilla* y haga click en el botón Continuar.

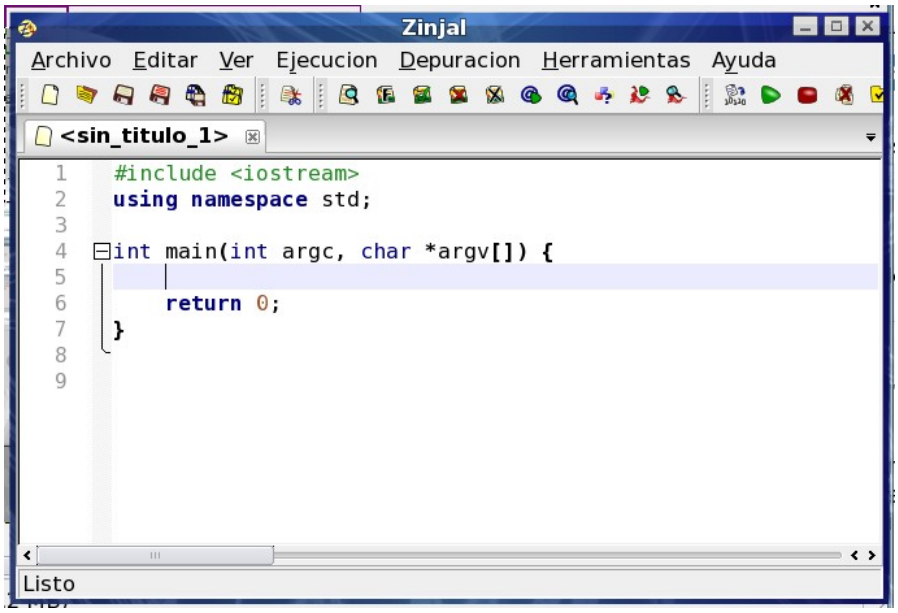


A continuación seleccione la plantilla Programa *C++ en Blanco* y presione *Crear*.



Esta acción cerrará el asistente y le abrirá una nueva pestaña con el esqueleto de un programa C++ (incluye la cabecera *iostream* y define la función *main*), y le posicionará el cursor en la primer línea de la función *main* para que comience a escribir[2].

2 Puede configurar ZinjaI para que al presionar Ctrl+N se cree automáticamente un nuevo programa utilizando la plantilla por defecto sin mostrar ninguno de los cuadros del asistente.



Paso 2:

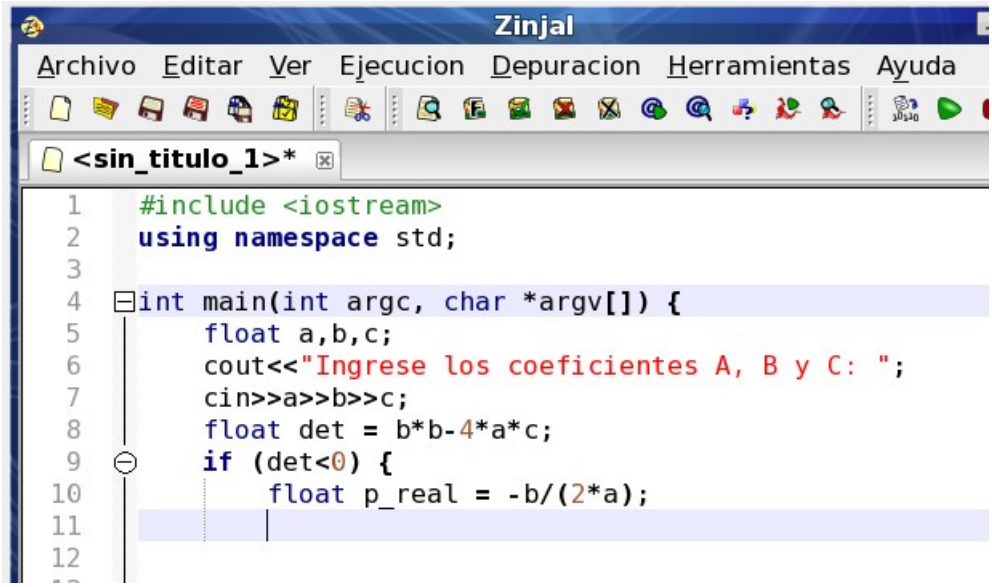
Lo siguiente que haremos será escribir el programa. Una forma de resolver el ejercicio se presenta a continuación:

```
#include <iostream>
using namespace std;

int main(int argc, char *argv[]) {
   float a,b,c;
   cout<<"Ingrese los coeficientes A, B y C: ";
   cin>>a>>b>>c;
   float det = b*b-4*a*c;
   if (det<0) {
      float p_real = -b/(2*a);
      float p_imag = sqrt(-det)/(2*a);
      cout<<"Las raices son: "
          <<p_real<<"+"<<p_imag<<"i y "
          <<p_real<<"-"<<p_imag<<"i"<<endl;
   } else {
      float raiz1 = (-b+sqrt(det))/(2*a);
      float raiz2 = (-b-sqrt(det))/(2*a);
      cout<<"Las raices son: "<<raiz1<<" y “
          <<raiz2<<endl;
   }
   return 0;
}
```

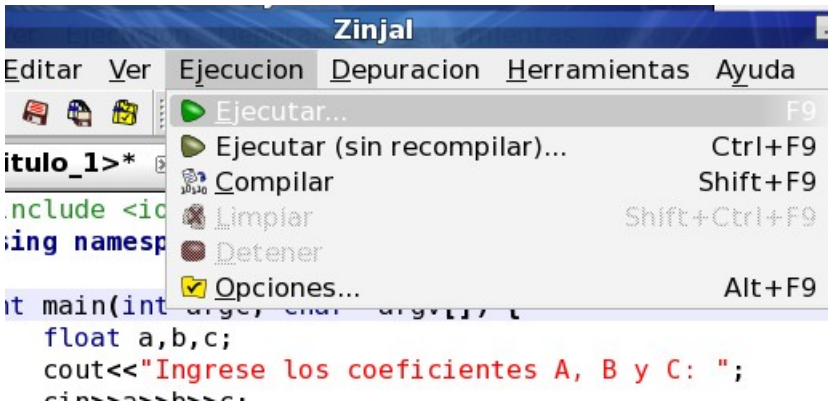
Mientras escribe, no se preocupe por el sangrado de las lineas, observe como *Zinjal* acomoda el cursor en el lugar adecuado cada vez que presiona *enter*[3].

3 Indentar el código es una costumbre altamente recomendable, ya que facilita enormemente la legibilidad del mismo.

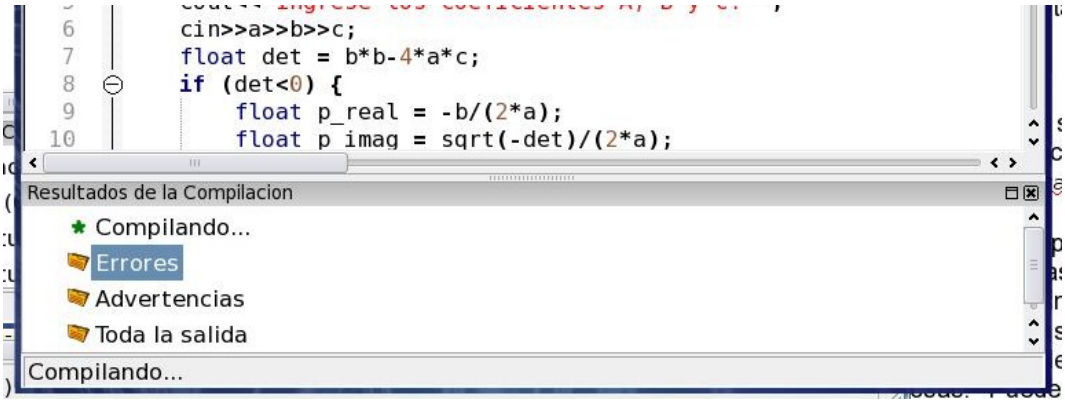


Paso 3:

Para intentar ejecutar el programa presione *F9*, o seleccione la opción *Ejecutar* del menú *Ejecutar*.



Esta acción guarda el archivo (si aún no tiene nombre la hará en un directorio temporal), lo compila, y si la compilación es exitosa lo ejecuta. Aparecerá en la parte inferior de la ventana principal el Panel de Resultados del Compilador, en el cual se muestra el estado de la compilación y los resultados de la misma.

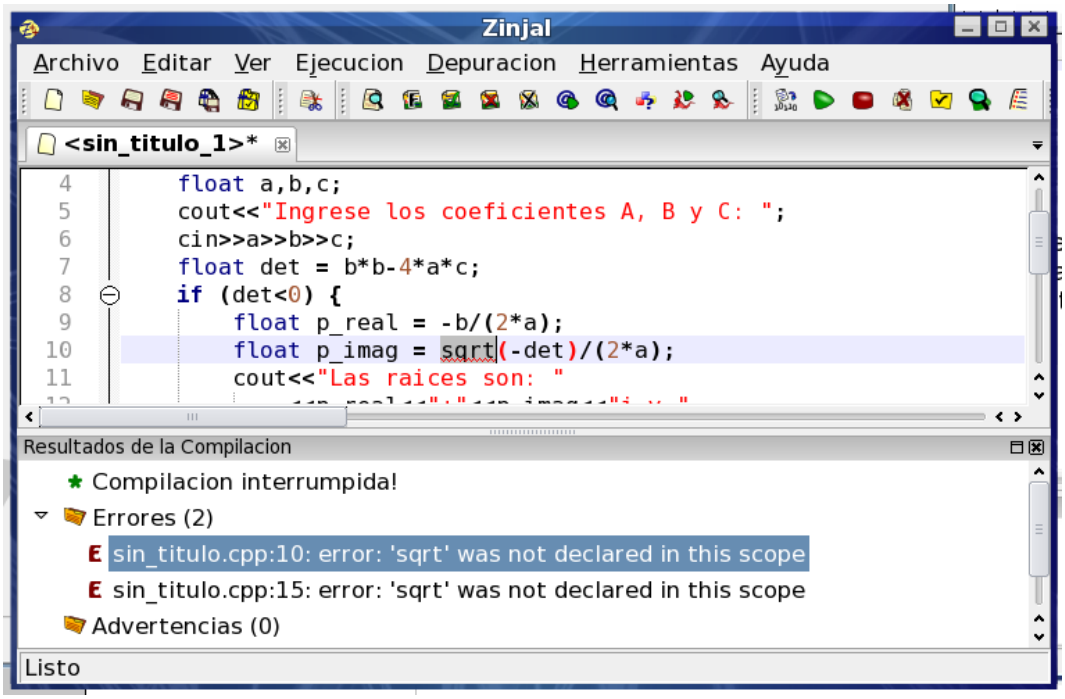


Debido a que el código copiado contiene un error (no se ha incluido la librería *cmath* para poder utilizar la función sqrt), el árbol de dicho panel desplegará la sección *Errores* mostrando un error similar a `"'sqrt' was not declared in this scope"` ('sqrt' no está declarado en este ámbito).

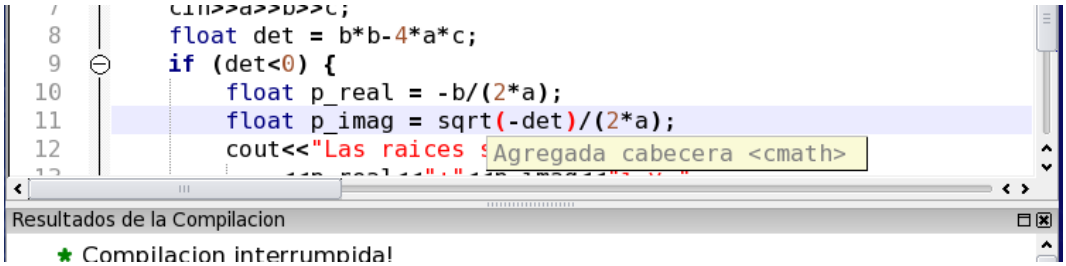
Notar que además de errores (de sintaxis), pueden aparecer advertencias en el panel de resultados de la compilación. Las advertencias (*warnings*) no impiden la compilación del programa (ya que la sintaxis no es incorrecta), pero indican posibles fuentes de error (como usar una variable sin inicializarla), malas prácticas (como no cumplir parcialmente el estándar, o declarar variables que no se utilizan), u otras construcciones dudosas. Pueden resultar útiles para encontrar errores de lógica.

Paso 4:

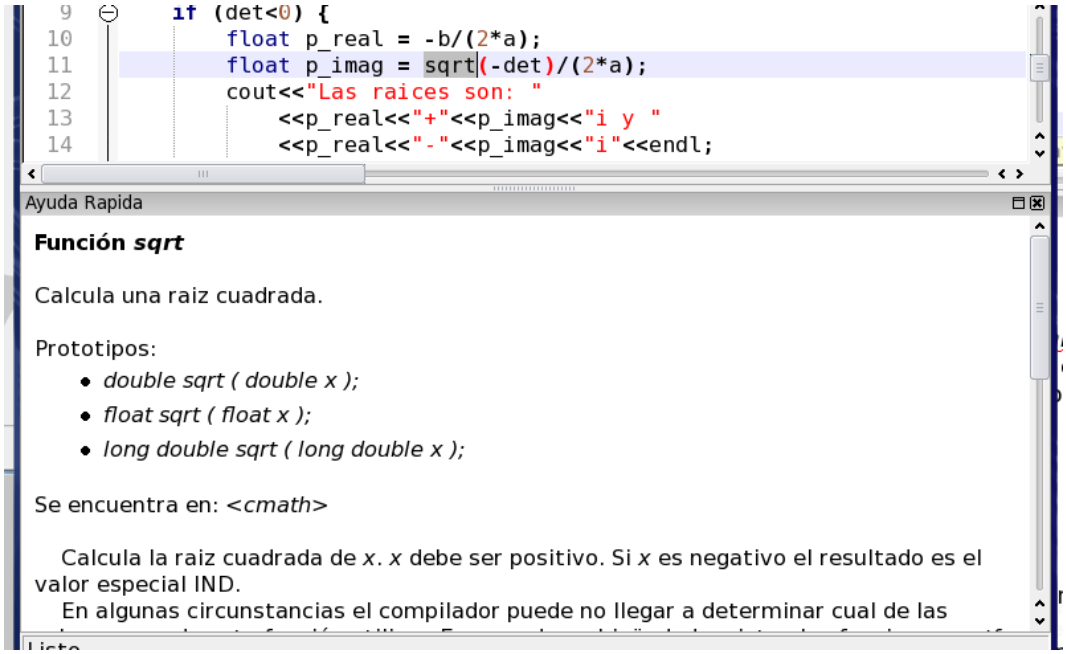
Haga doble click sobre el error en el panel de compilación y observe como en el editor el cursor se desplaza hacia la linea que lo provocó, y subraya con rojo la función a la cual el mensaje de error hace referencia.



Para solucionar el error debería incluir la cabecera cmath al principio del archivo. *Zinjal* puede hacer esto automáticamente. Presione *Ctrl+H* y observe como *Zinjal* añade la linea.

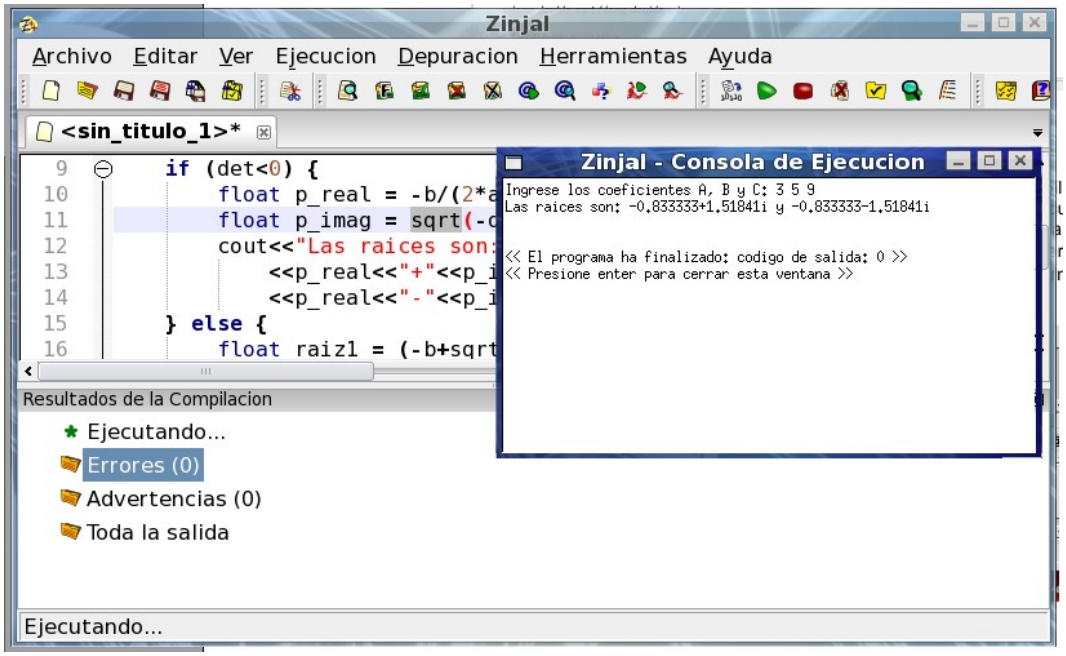


Otra forma de resolver el problema sería presionar *Shift+F1* para abrir el panel de ayuda rápida, el cual brinda información sobre los elementos estándar del lenguaje, así como ejemplos de uso y enlaces a elementos relacionados.



Paso 5:

Presione nuevamente *F9* para correr el programa. Esta vez se compilará y ejecutará correctamente en una nueva ventana. Luego de finalizar la ejecución, *Zinjal* informará el código de retorno de su programa (el 0 de la línea final "`return 0;`", el cual sirve para saber si se ejecutó correctamente) y esperará a que presione *enter* una vez más antes de cerrar la ventana, para permitirle observar los resultados.



Otras consideraciones

Atajos de teclado de utilidad

Es importante que el programador se sienta cómodo con las facilidades de edición que le brinda la IDE que utiliza. Conocer los atajos de teclado para acciones muy frecuentes aumenta mucho la velocidad de escritura del código evitando destinar tiempo a tareas tediosas o repetitivas que nada tienen que ver con la lógica del problema.

A continuación se listan algunas combinaciones de teclas para aprovechar mejor algunas facilidades de edición Zinjal:

- F9: Este atajo realiza todos los pasos necesarios para probar un programa (guardar, compilar y ejecutar). Si se presiona Shift+F9, se evita el último paso; es decir, sólo se compila. Esto sirve para saber si el código es sintácticamente correcto.
- Ctrl+<: Si la compilación arroja errores o advertencias, con esta combinación se pueden recorrer los mismos. Al utilizarla, se selecciona un error y el cursor se posiciona en la línea que ocasionó el mismo. El error que se selecciona va variando en cada pulsación.
- Ctrl+H: Como se vio en el ejemplo, esta combinación busca la cabecera que contiene la declaración de una determinada clase, función, variable o macro e inserta al principio del archivo el #include que corresponda para poder utilizarla. La palabra que se busca es siempre la seleccionada o la sobre la cual está el cursor de texto.
- Ctrl+L, Ctrl+Shift+L: La primera duplica la linea actual o las líneas seleccionadas. Es útil en muchos casos en que el código incluye líneas casi idénticas (por ejemplo, en el cálculo de las dos raices del ejemplo anterior), equivale a copiar y pegar esas líneas. La segunda combinación elimina la línea actual o las líneas seleccionadas.
- Ctrl+T, Ctrl+Shift+T: estas combinaciones desplazan la línea actual o las líneas seleccionadas una posición más arriba en el código. Sirven para mover fragmentos de código lineas arriba o abajo.

- Ctrl+D, Ctrl+Shift+D: Estas combinaciones sirven para comentar/descomentar respectivamente una o más líneas.
- Ctrl+I: Este atajo corrige el indentado (los espacios al principio de cada línea) de un conjunto de líneas para facilitar la lectura del código.
- Shift+F1: Estando posicionado con el cursor de texto sobre una palabra clave o identificador, este atajo invoca al panel de ayuda rápida presentando un texto de ayuda relacionado al mismo.

Depuración con Zinjal

Casi todos los IDEs existentes integran en mayor o menor grado facilidades para la depuración. Esto es, permiten interrumpir la ejecución de un programa para evaluar variables o expresiones, modificarlas, observar las llamadas a funciones realizadas, continuar paso por paso, etc.

Para iniciar una sesión de depuración en Zinjal se debe ejecutar el programa de forma especial, con la tecla F5 en lugar de F9. Todo lo que se puede hacer durante la depuración está disponible en el menú “Depuración” o en los paneles que aparecen al presionar F5. Si un alumno posee experiencia en la utilización de herramientas de depuración con otras IDEs, puede explorar esta opciones. En cualquier caso, una descripción más detallada de las posibilidades y cómo aprovecharlas será motivo de otro anexo cuando el alumno se encuentre más familiarizado con lo básico presentado anteriormente y el lenguaje.