



ASRIEL

flambo

DiracSpace



ASRIEL

Descripción General

Este es un programa que permite emular código del lenguaje C++, así como generar y editar archivos.

Metas

Ser una plataforma de desarrollo mejor conocida; para así ser implementada por desarrolladores que puedan acceder a ella y editarla si así lo desean.

Todo sin fines de lucro.

Requerimientos

MinGW-W64

Tenerlo instalado y agregado a tus variables de entorno.

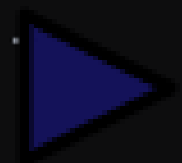
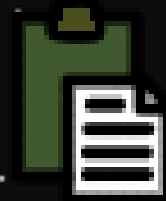
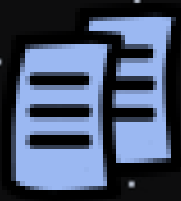
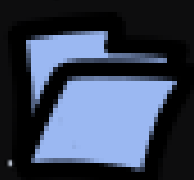
Java Runtime Environment

Tenerlo instalado en tu máquina.

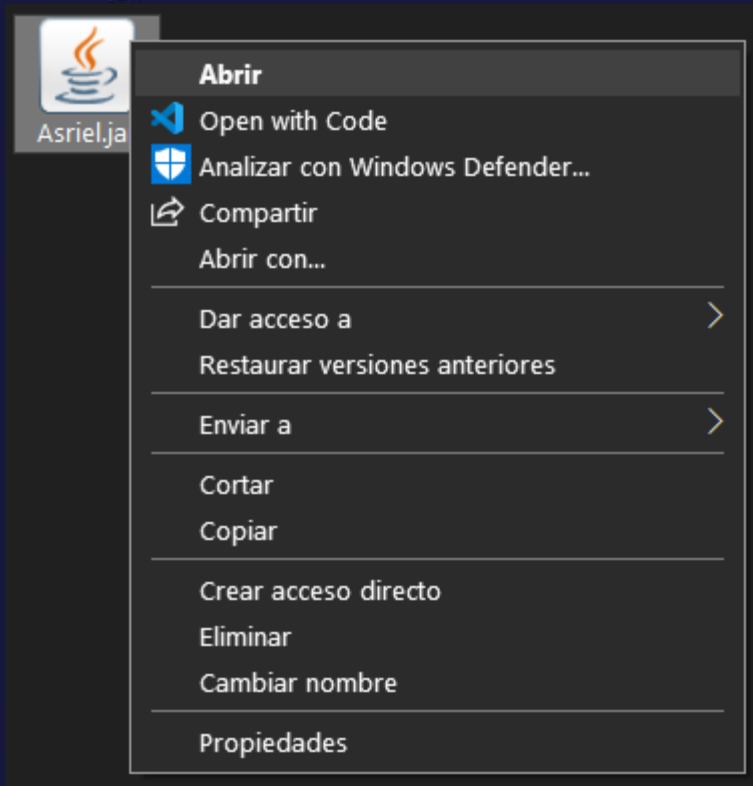


ASRIEL

```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      cout << "Manual De Usuario" << endl;
6      return 0;
7  }
```



1. Ejecutar la aplicación

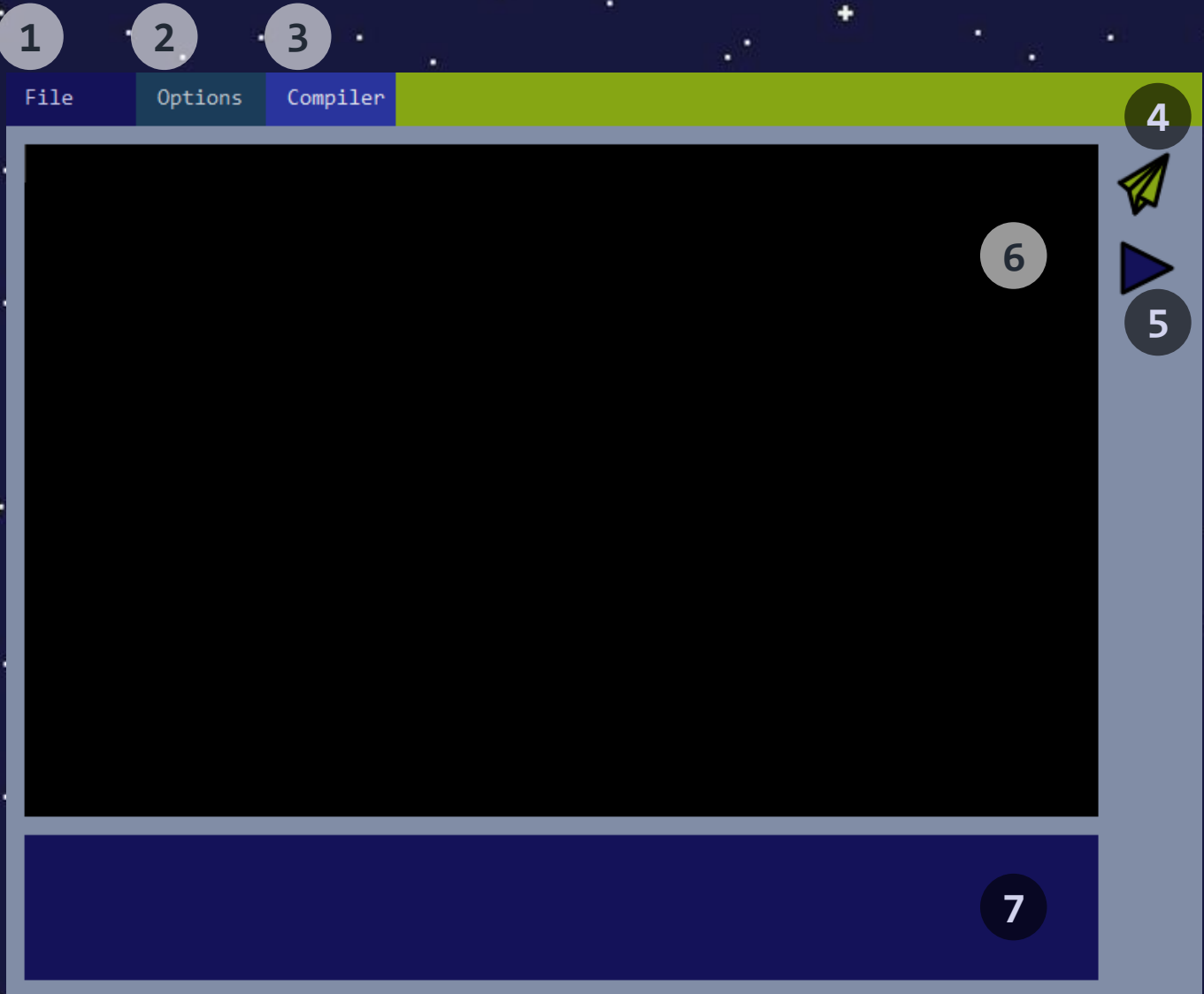


Para ello debes dar clic derecho sobre el icono de la aplicación, y dar clic en la opción ejecutar o abrir.

Te saldrá esta imagen de carga, que dependiendo de la velocidad de ejecución del dispositivo que uses, tardara en emular el compilador.



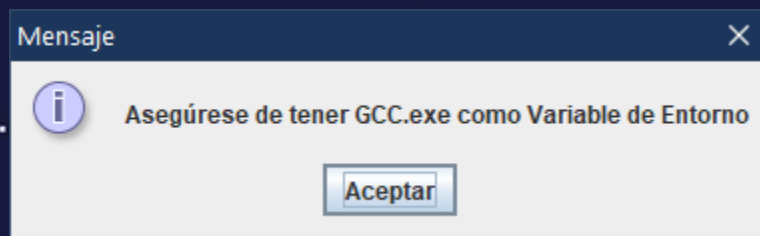
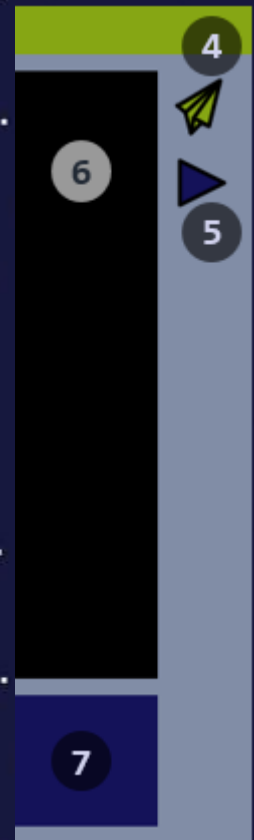
2. Componentes



Una vez cargado Asriel, se verá así.

1. Es un menú desplegable que permite el manejo de archivos.
2. Es un menú desplegable que permite el manejo de elementos de texto sobre el portapapeles.
3. Es un menú de texto que permite el generar código intermedio y ensamblador.

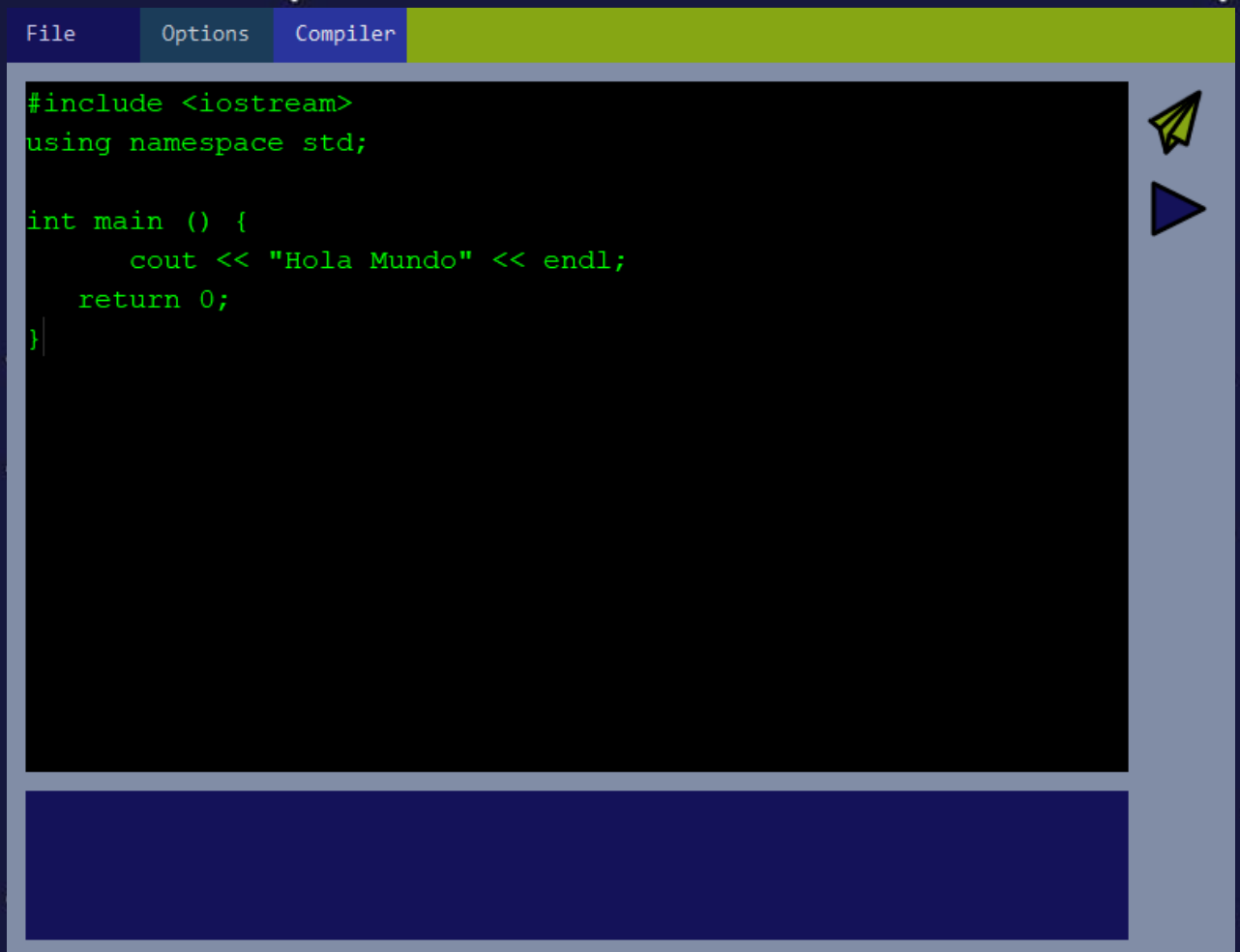
4. Este es un botón que te permite compilar el código.
5. Este es un botón que te permite ejecutar el código.
6. Esta es la ventana donde se escribe o muestra el código que se va a manejar.
7. Este recuadro muestra si el código ha sido correctamente emulado o no.



Pd. Esta ventana te saldrá como advertencia cada que ejecutes la aplicación. Es para prevenir cualquier fallo.

3. Hola mundo

Una vez que ya tienes la app en funcionamiento y conoces sus partes vamos a llevar un hola mundo. Toma en cuenta que este compilador va enfocado al lenguaje c++, así que considera sus reglas.

A screenshot of a C++ IDE window. The window has a title bar with three tabs: 'File', 'Options', and 'Compiler'. The 'Compiler' tab is active. The main editor area is dark and contains the following C++ code:

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

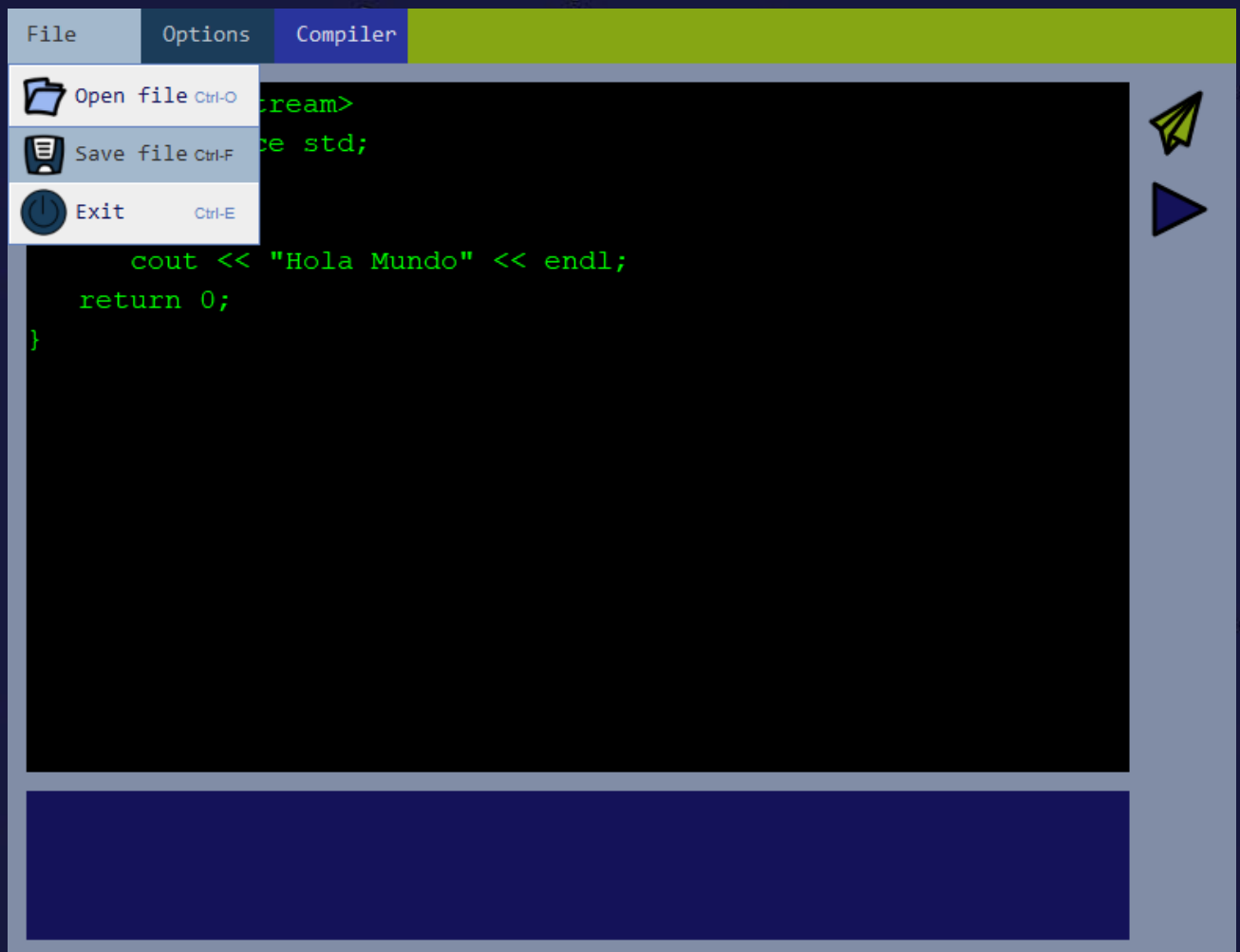
int main () {
    cout << "Hola Mundo" << endl;
    return 0;
}
```

On the right side of the editor, there are two icons: a yellow paper airplane and a blue triangle. Below the editor is a large, empty blue rectangular area, likely for output or debugging.

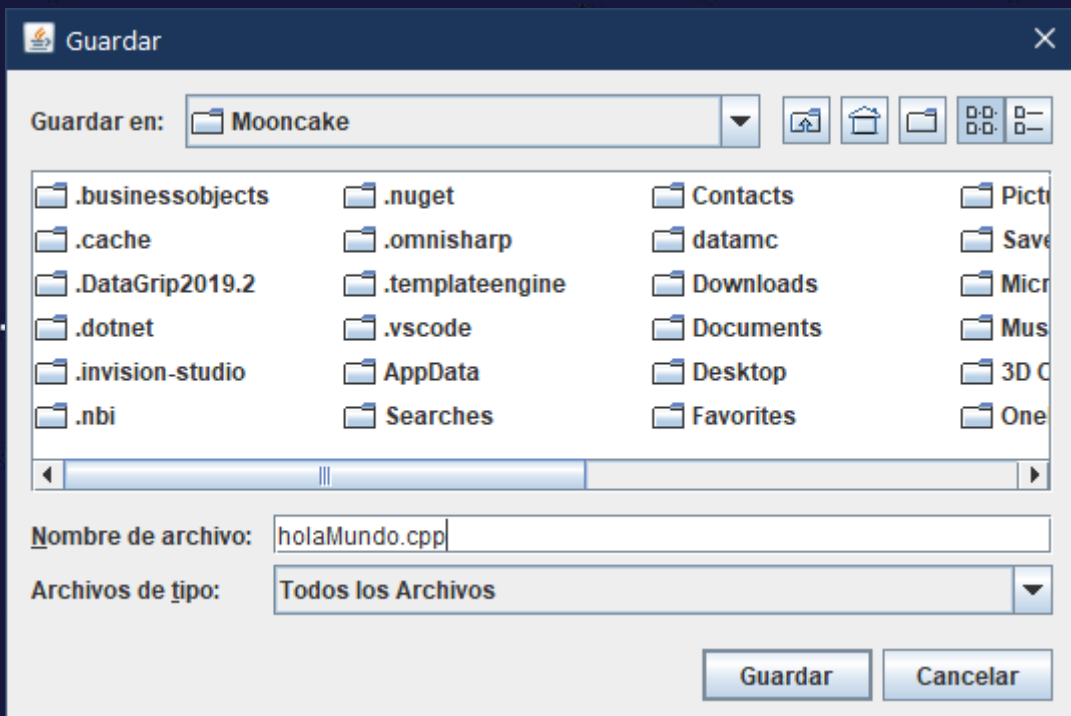
En la imagen anterior, vimos un ejemplo sencillo de como se realiza una impresión de texto con el clásico 'Hola mundo', pero tu puedes hacer la del algoritmo que desees probar siempre y cuando este dentro de las normas de codificación del lenguaje c++.



Cuando tengas el código listo es hora de llevarlo a ejecución. Para ello debemos guardarlo primero.



Cuando hayas dado clic en la opción 'Save file', te saldrá esta ventana, en donde solo debes dar el nombre (No olvides la extensión .cpp) y escoger donde es que quieres que se guarde el algoritmo.



Una vez que ya tengas el documento guardado en tu equipo ahora puedes compilarlo y ejecutarlo sin ningún problema.



Primero dando clic en el botón de compilar.

Si tu código fue escrito correctamente sale este mensaje.

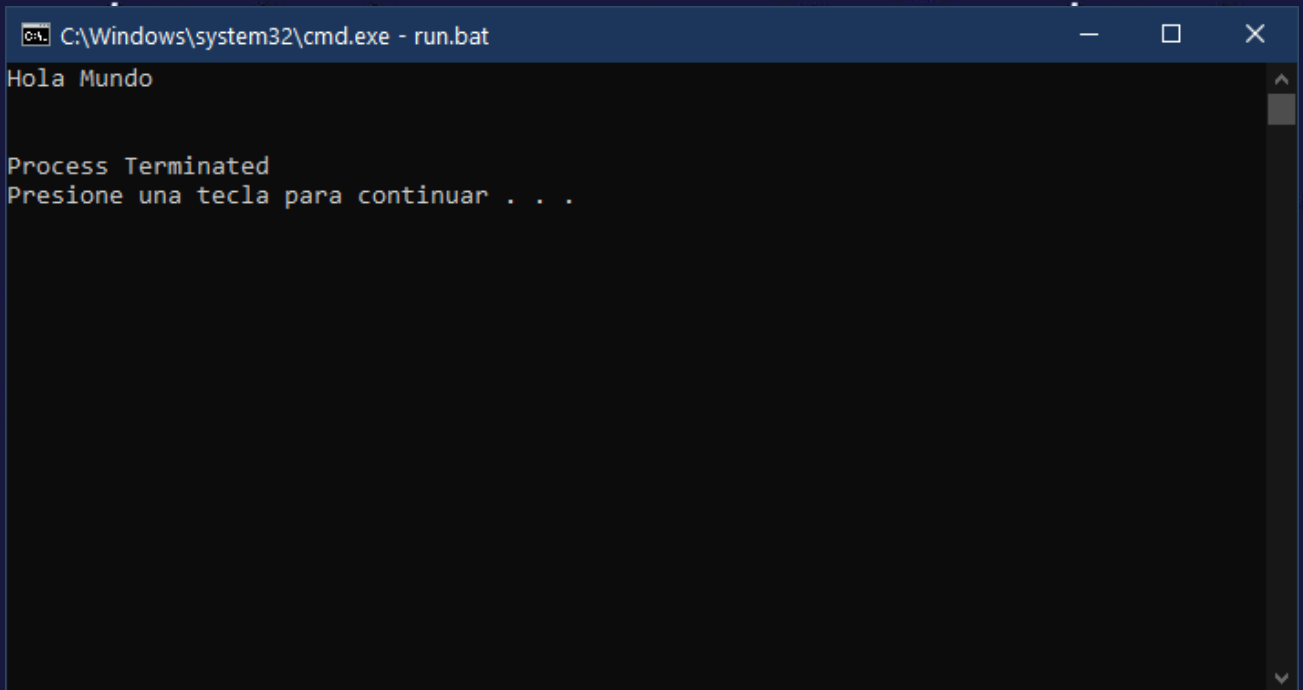
```
== 0 error.. Compilation Finished
```

Una vez que hayas compilado el código, debes ejecutarlo.

Para ejecutarlo debes dar clic en el botón de ejecutar.



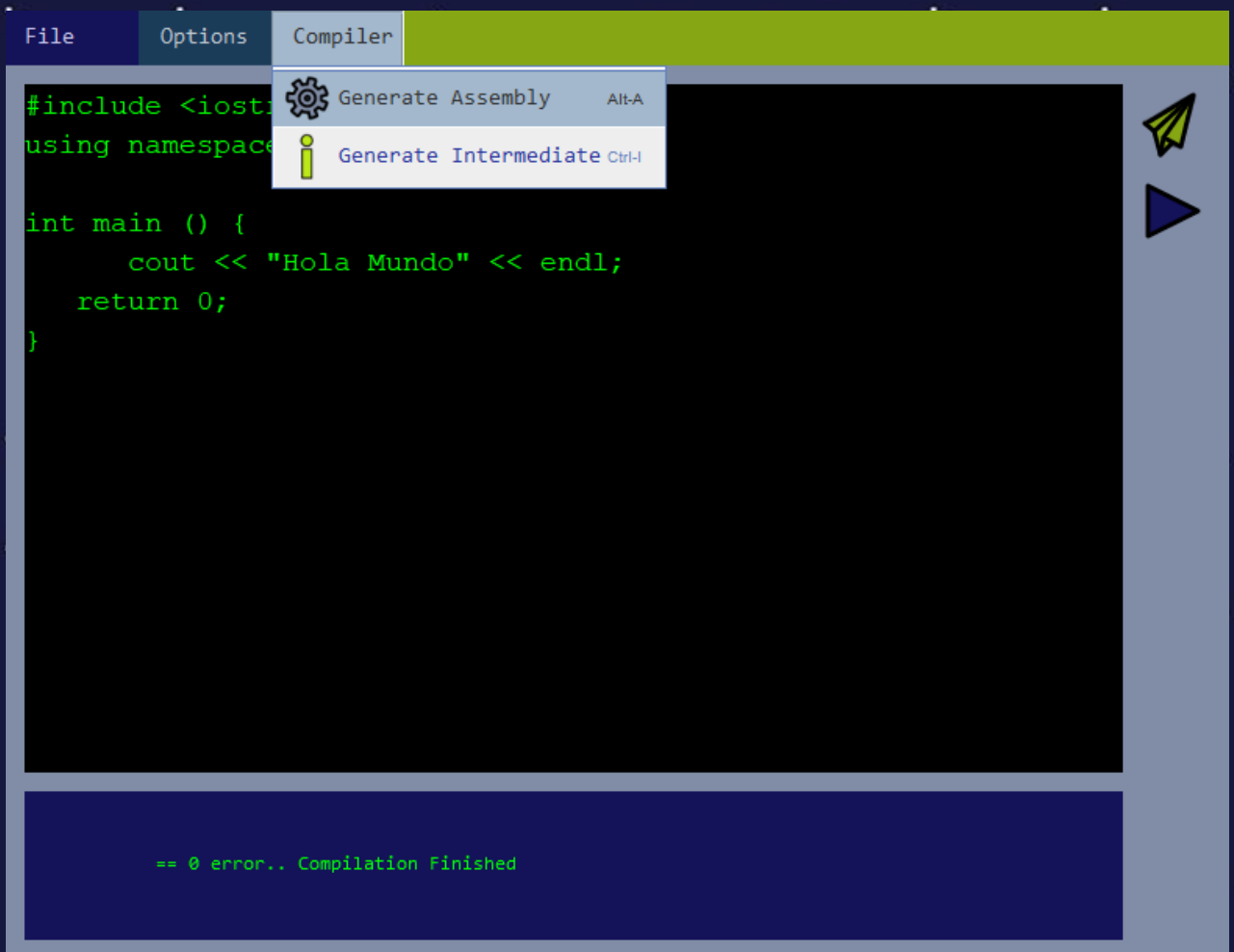
Esto te arrojará esta ventana que es el CMD, donde te muestra el resultado de tu algoritmo.

A screenshot of a Windows Command Prompt window. The title bar reads "C:\Windows\system32\cmd.exe - run.bat". The window content shows "Hola Mundo" in green text, followed by "Process Terminated" and "Presione una tecla para continuar . . ." in white text. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Una vez que des un clic en alguna tecla de tu teclado (valga la redundancia), se te mostrará nuevamente la ventana de Asriel.

4. Lenguaje ensamblador

Cuando tienes tu código funcional, ya lo compilaste y ejecutaste con éxito. Es posible generar su ensamblador. Para ello nos vamos a la pestaña 'Compiler'.

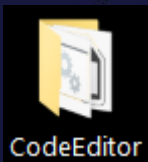


Aquí nos vienen dos opciones, por ahora escogeremos la que dice 'Generate Assembly'.

Se te muestra este mensaje, que te afirma que ha sido realizado con éxito el código ensamblador así como su dirección.

```
Assembly generated in -> C:\Users\Mooncake\Desktop\CodeEditor\holaMundo.asm
```

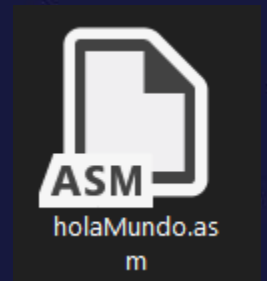
Si te vas a tu escritorio encontraras una carpeta que se llama 'CodeEditor'.



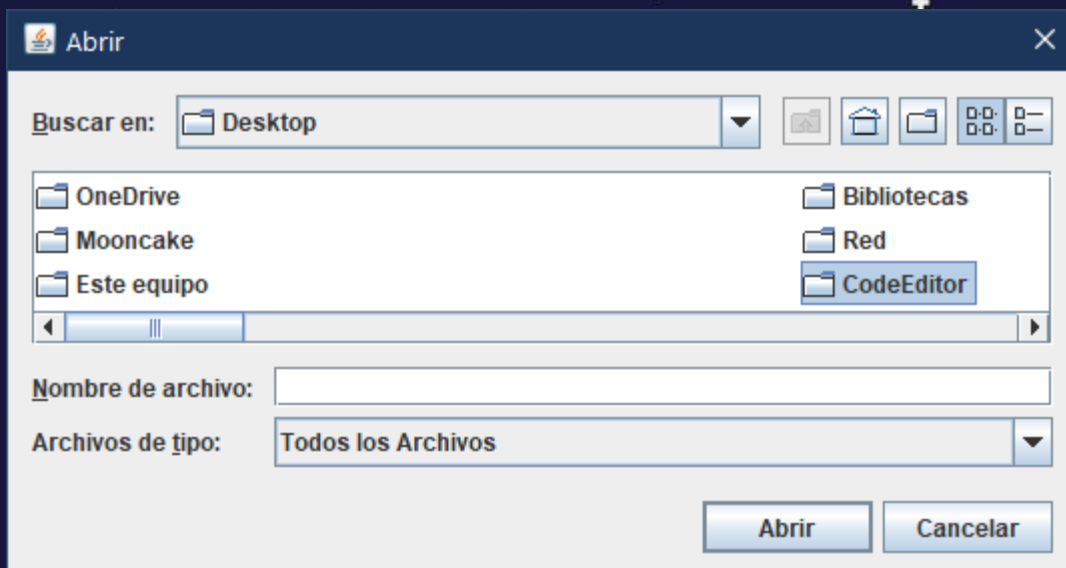
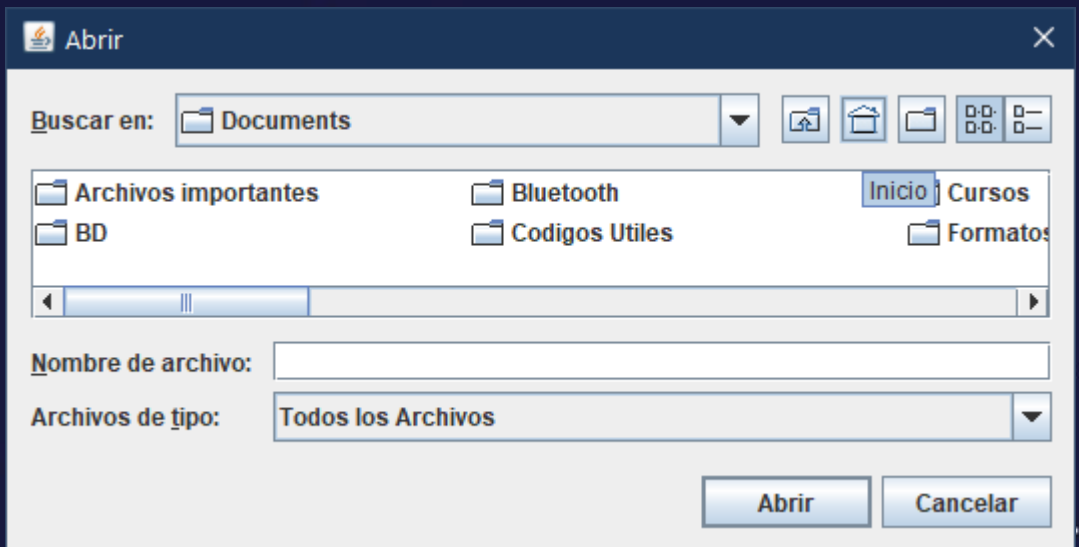
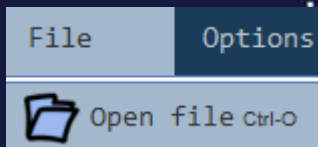
Dentro de la cual se guardaran todos los archivos ensambladores (.asm) que generes.

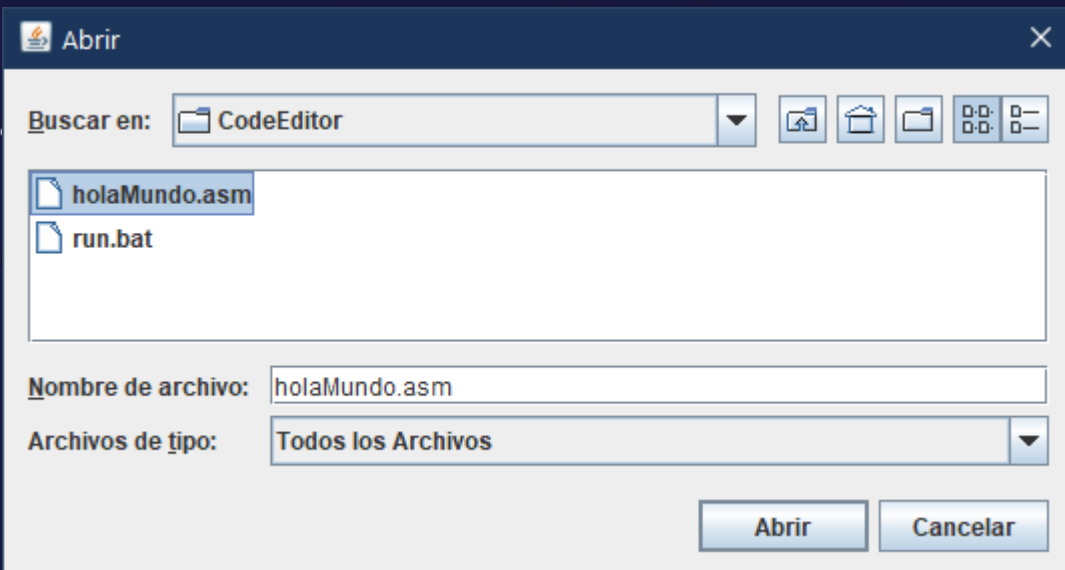
Esta carpeta es generada en tu escritorio a la par que ejecutas por primera vez Asriel. Si quieres ver tu archivo ensamblador. Solo debes ir ahí y buscarlo por el nombre del algoritmo que emulaste. En este caso 'holaMundo.asm'

Este es el lenguaje ensamblador que se genero a partir de nuestro algoritmo 'holaMundo'.

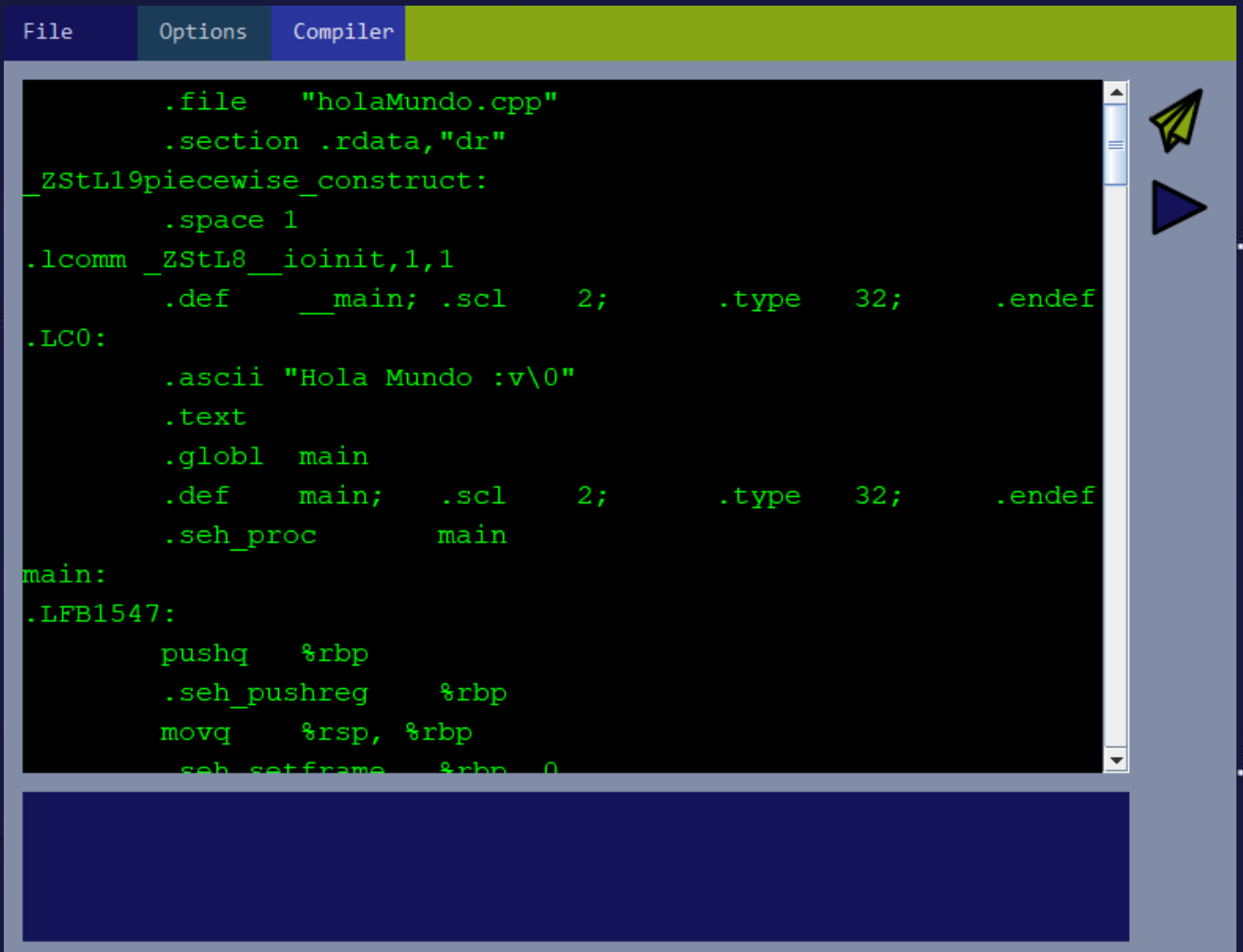


Si tienes curiosidad de saber como se, puedes abrirlo con Asriel. Solo debes irte a la pestaña 'File' y luego en la opción 'Open file'. Y estando en el buscador, solo te vas a escritorio, luego a la carpeta 'CodeEditor' y estando ahí solo resta escoger el archivo ensamblador correspondiente a tu algoritmo.





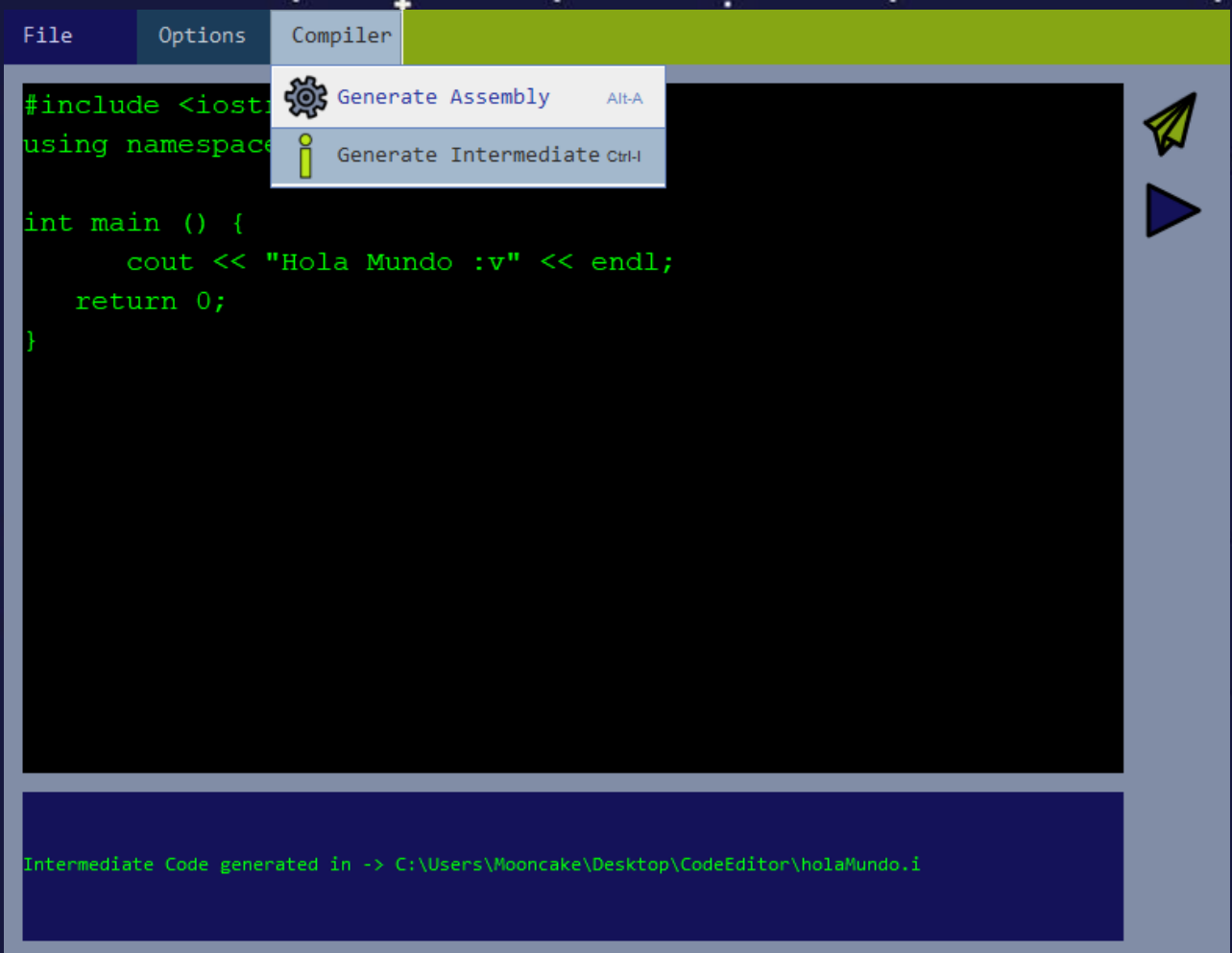
4



5

5. Código intermedio

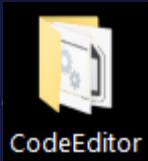
Para generarlo debemos hacer lo mismo que con el ensamblador, solo que ahora escogemos la opción 'Generate Intermediate'.



Del mismo modo nos muestra un mensaje que nos afirma que se ha generado con éxito.

Intermediate Code generated in -> C:\Users\Mooncake\Desktop\CodeEditor\holaMundo.i

Si deseas saber donde esta, solo debes ir a tu escritorio, en la misma carpeta 'CodeEditor'.



Dentro de la cual se guardaran al igual que los archivos ensambladores (.asm), todos los códigos intermedios (.ii) que generes.

Solo debes ir ahí y buscar el código por el nombre del algoritmo que emulaste. En este caso es 'holaMundo.ii'

Este es el código intermedio que se genero a partir de nuestro algoritmo 'holaMundo'.

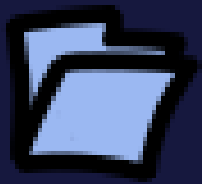


holaMundo.ii

Si tienes curiosidad de saber como se, puedes abrirlo con Asriel. Solo debes irte a la pestaña 'File' y luego en la opción 'Open file'. Y estando en el buscador, solo te vas a escritorio, luego a la carpeta 'CodeEditor' y estando ahí solo resta escoger el archivo ensamblador correspondiente a tu algoritmo. Esto, siguiendo los pasos anteriormente marcados.

6. Atajos de teclado

Para hacer todo de un modo más práctico, las acciones realizadas desde la barra de menú tienen atajos que te permitirán trabajar de forma mucho más rápida y sencilla.



Abrir
archivos
Ctrl+O



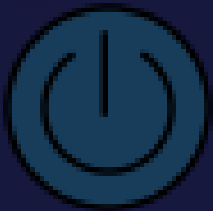
Generar código
intermedio
Ctrl+I



Guardar
archivos
Ctrl+F



Generar lenguaje
de ensamblador
Alt+A



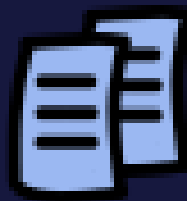
Cerrar
Asriel
Ctrl+E



Cortar
texto
Ctrl+C



Copiar
todo
Ctrl+A



Pegar
texto
Ctrl+P

Para mayor información

```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      cout << "correo de DiracSpace: ralphlaurenpolo@gmail.com" << endl;
6      cout << "correo de flambo: spiderman0698@gmail.com" << endl;
7      return 0;
8  }
```

¡Recuerda
apoyar el
Open
Source!



El link de
descarga del
código de
Asriel está en
el QR

