

# PyTiger2C

## Documentación de la interfaz gráfica

Yasser González Fernández  
yglez@uh.cu

Ariel Hernández Amador  
gnuaha7@uh.cu

Este documento muestra, de manera general, los pasos a seguir para crear un nuevo programa Tiger utilizando la interfaz gráfica de PyTiger2C. Además, ilustra otras funcionalidades de la interfaz gráfica como la posibilidad de mostrar el árbol de sintáxis abstracta correspondiente al programa y el código C generado.

La interfaz gráfica puede iniciarse ejecutando el *script* Python `gpytiger2c.py` que se encuentra en el directorio `scripts` de la distribución en código fuente de PyTiger2C.

Al ejecutar este *script*, se mostrará la ventana principal de la interfaz gráfica como se ilustra en la figura 1.

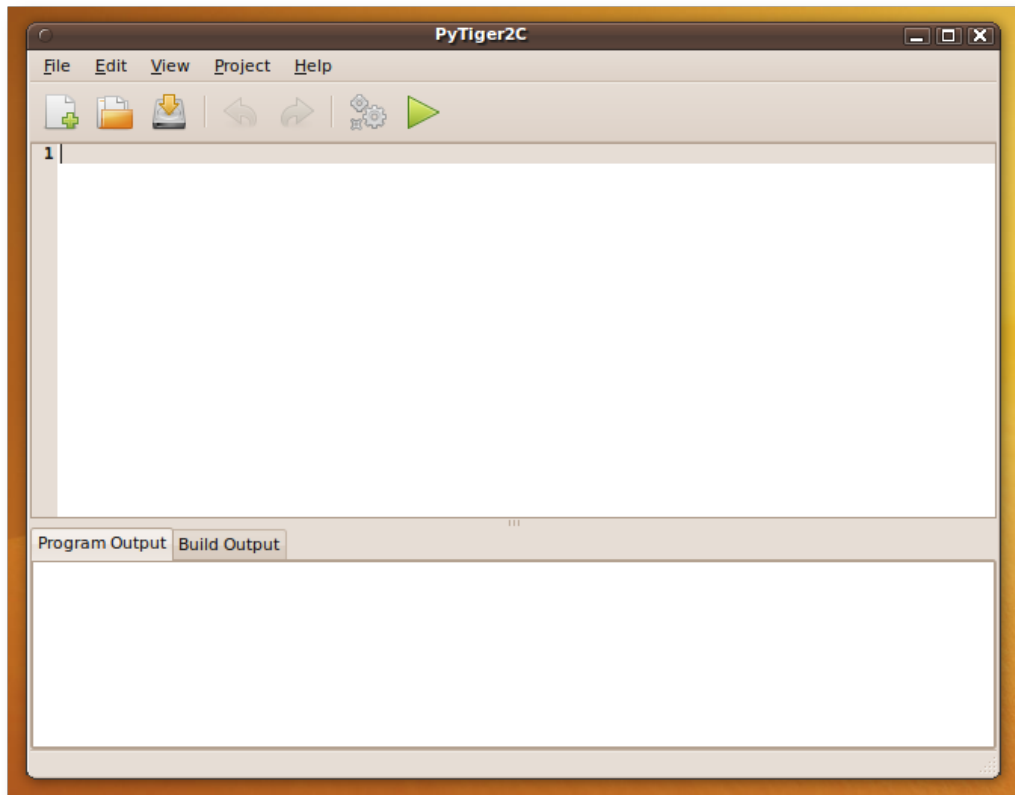


Figura 1: Ventana principal.

El elemento principal de esta ventana es el control *GtkSourceViewer* que permite introducir el código fuente del programa en el lenguaje Tiger. Este control brinda las siguientes funcionalidades: resaltado de sintáxis a partir de una descripción de la estructura del lenguaje mediante un archivo XML, permite deshacer y rehacer los cambios hechos durante la edición, muestra el número de cada línea del archivo y resalta la línea donde se encuentra el cursor y los caracteres que deben una pareja como paréntesis, llaves y corchetes.

Las funcionalidades disponibles durante la edición descritas anteriormente se ilustran en la figura 2 con una implementación del conocido programa *Hello, World!* en el lenguaje Tiger.

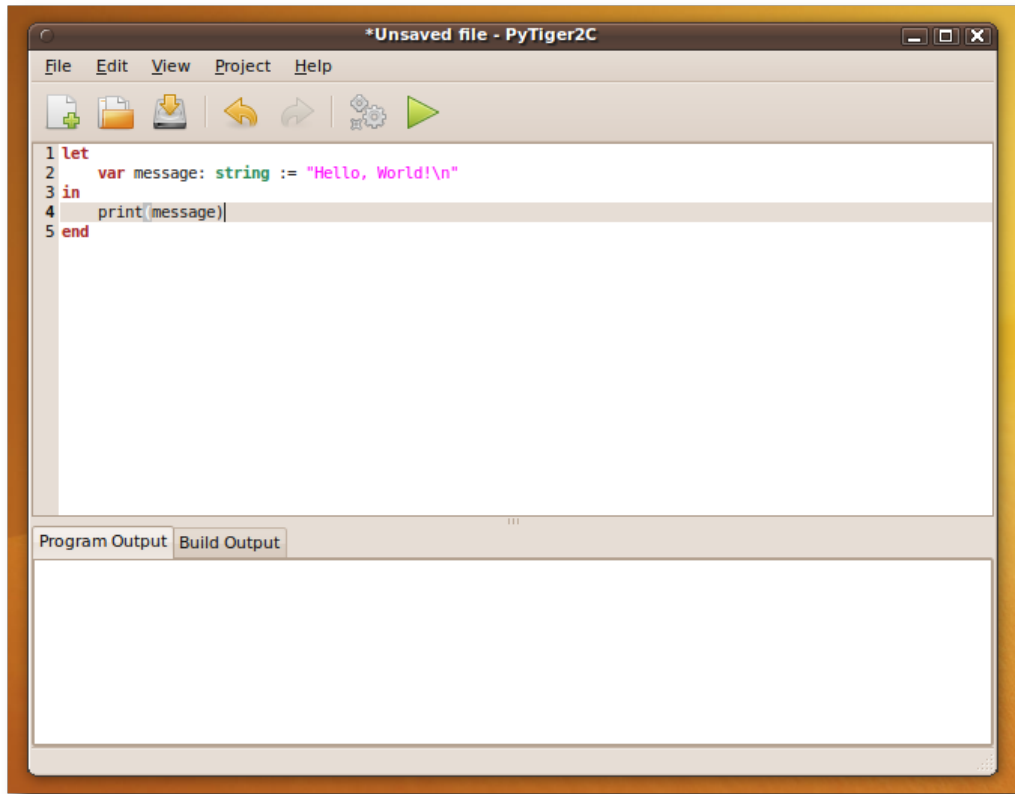


Figura 2: Creación de un programa.

Luego de terminada la edición del nuevo programa Tiger y antes de efectuar su compilación se debe guardar el programa en un nuevo archivo. Para hacer esto se puede utilizar el elemento *Save As...* del menú *File* como se ilustra en la figura 3.

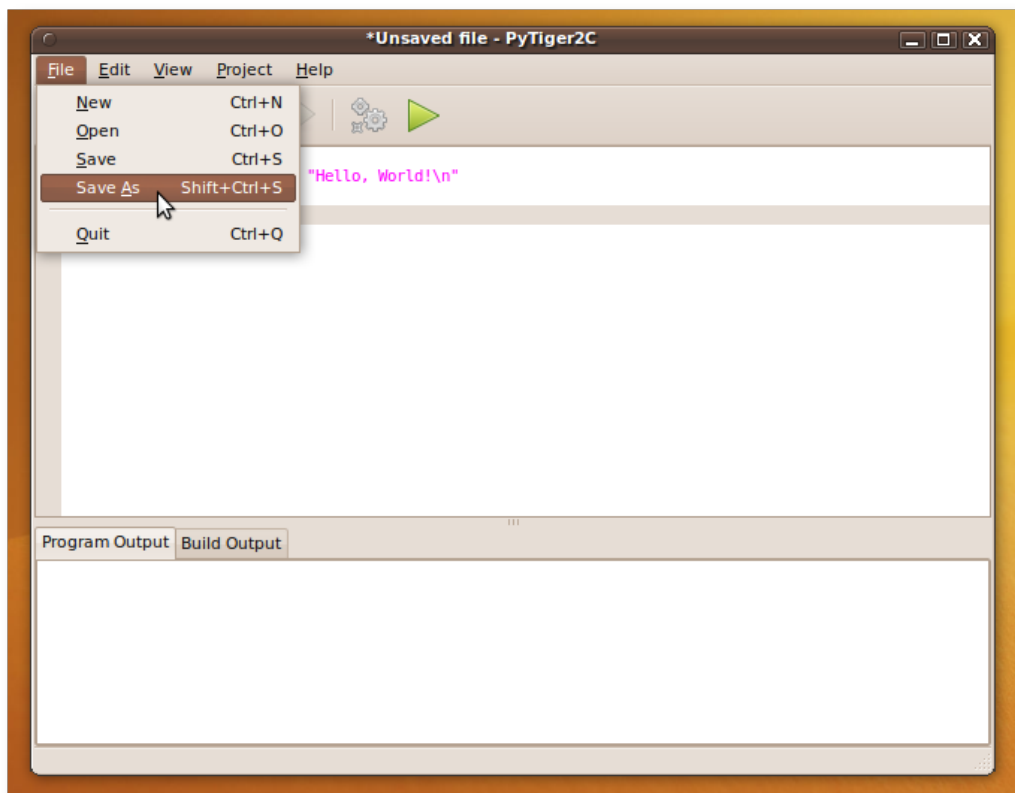


Figura 3: Guardando el nuevo programa creado.

La acción anterior mostrará un diálogo donde se debe especificar el nombre que deberá tener el archivo, el directorio donde se guardará y confirmar utilizando el botón *Save As* como se ilustra en la figura 4.

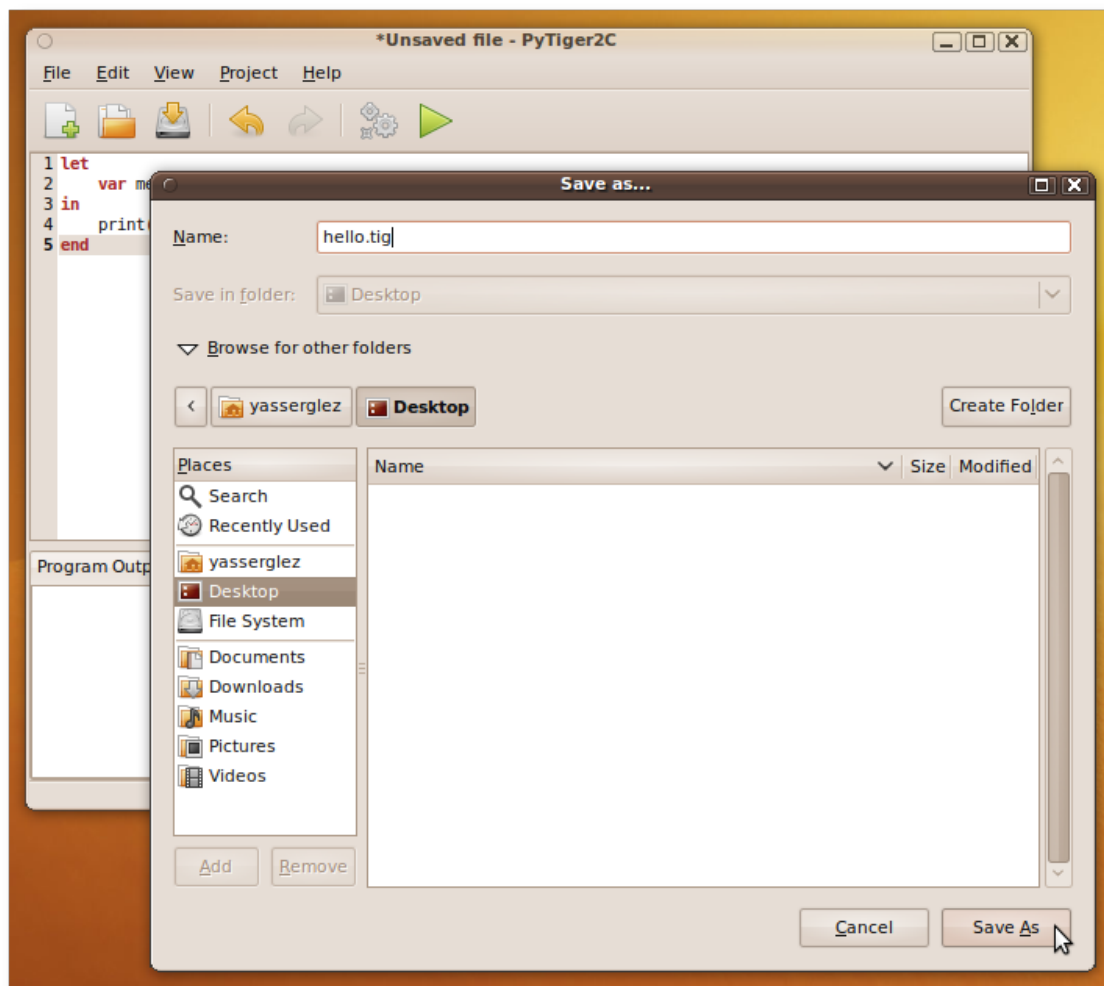


Figura 4: Introduciendo el nombre del nuevo archivo.

Una vez guardado el nuevo programa Tiger en un archivo es posible compilarlo utilizando el elemento *Build* del menú *Project* como se ilustra en la figura 5.

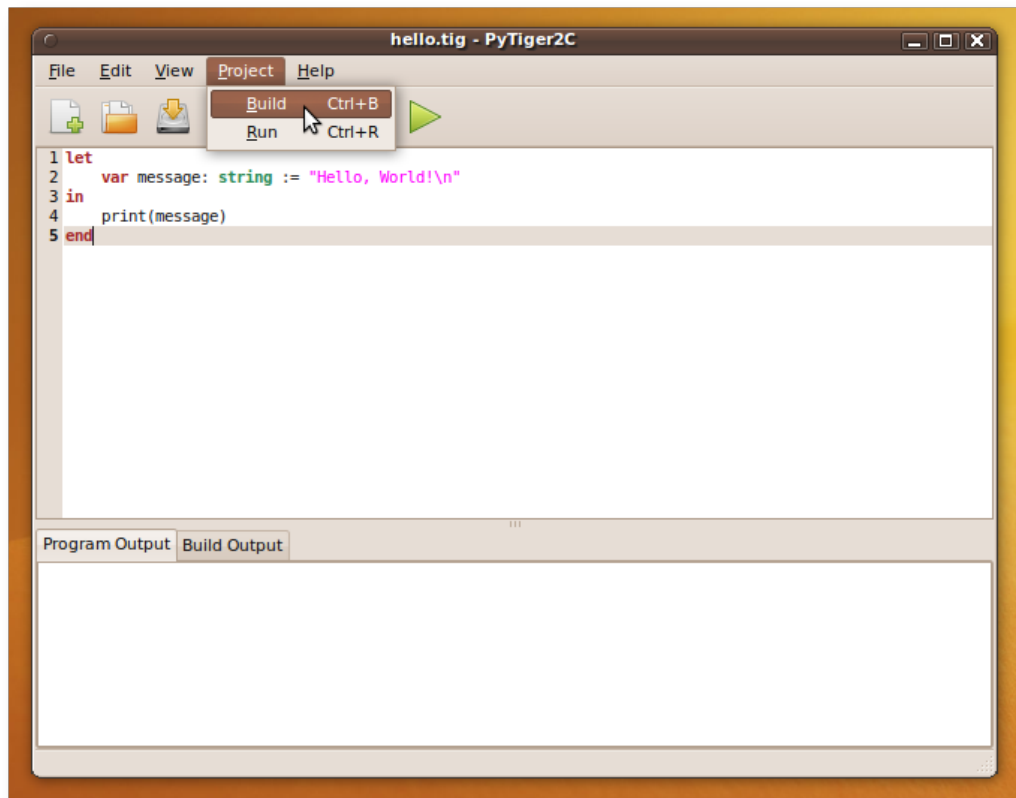


Figura 5: Compilando el nuevo programa.

Si el programa Tiger no tiene ningún error se mostrará, en la pestaña *Build Output*, un mensaje indicando que el proceso de compilación finalizó correctamente como se ilustra en la figura 6.

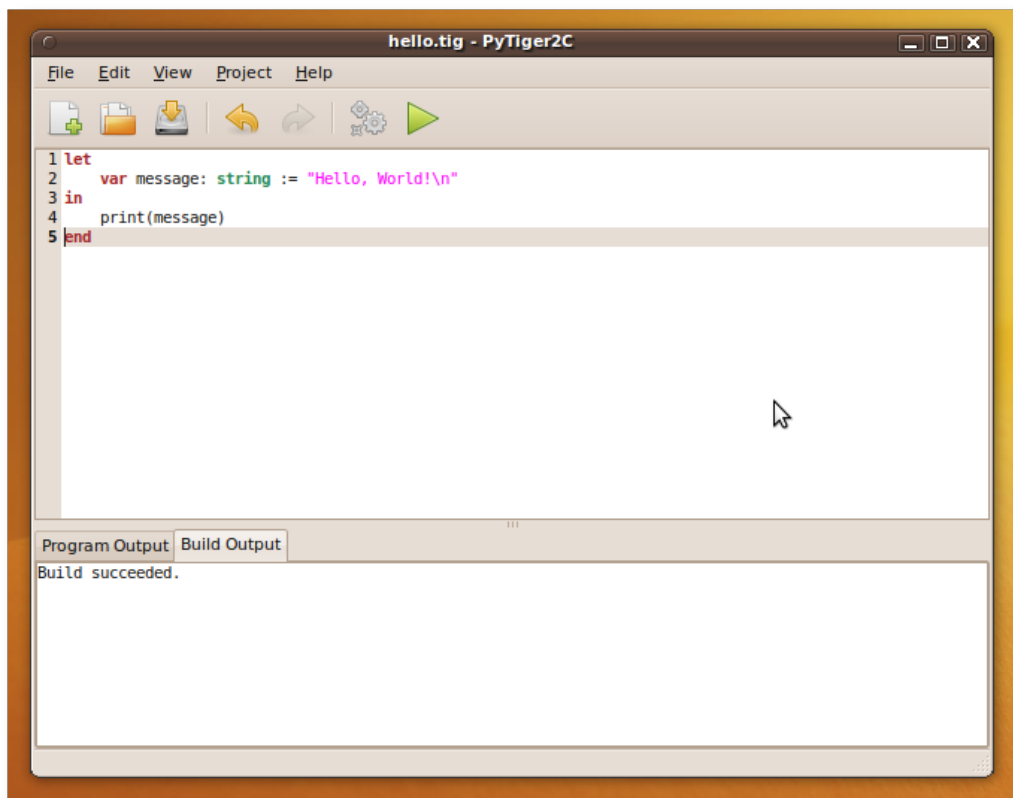


Figura 6: Mensaje indicando que el proceso de compilación finalizó correctamente.

Si el programa Tiger tuviera algún error semántico o sintáctico se mostrarán los mensajes correspondiente en la pestaña *Build Output*. Por ejemplo, la figura 7 muestra el mismo programa *Hello, World!* pero se ha sustituido el llamado a `print` de la línea 4 por un llamado a `printi`; esto genera un error semántico ya que la función `printi` recibe un `int` como argumento y se está llamando con un argumento `string`.

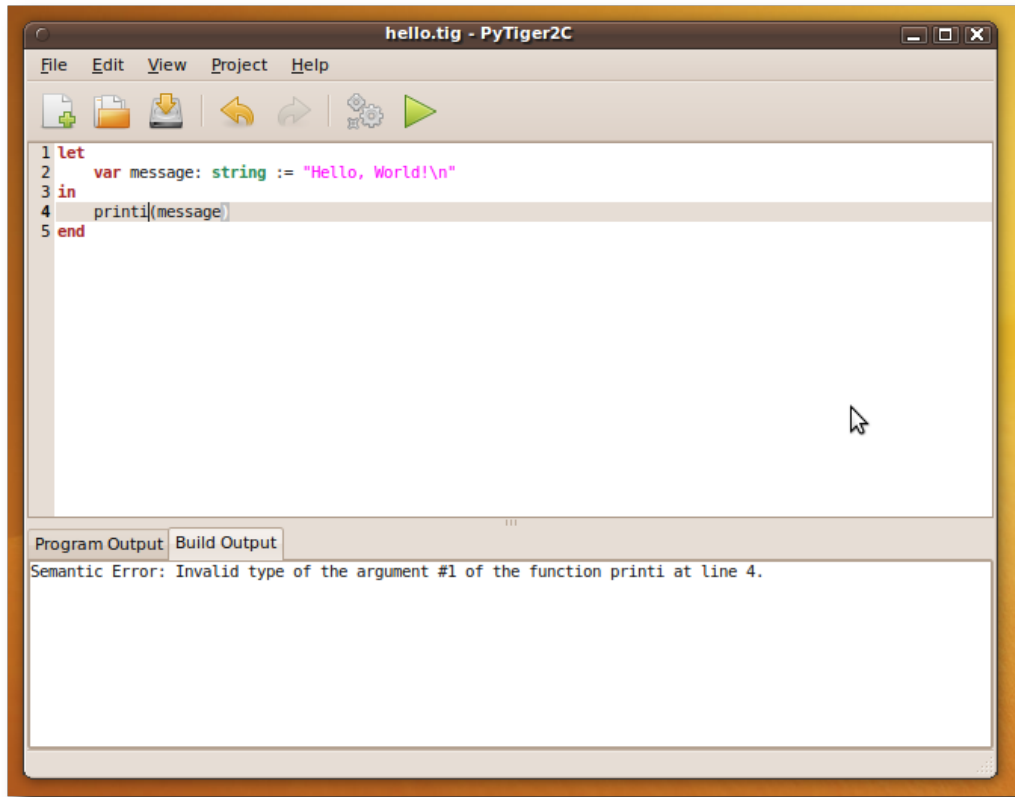


Figura 7: Mensaje indicando los errores presentes en el programa.



Una vez que el programa se ha compilado correctamente es posible ejecutarlo desde la interfaz gráfica utilizando el elemento *Run* del menú *Project* como se ilustra en la figura 8.

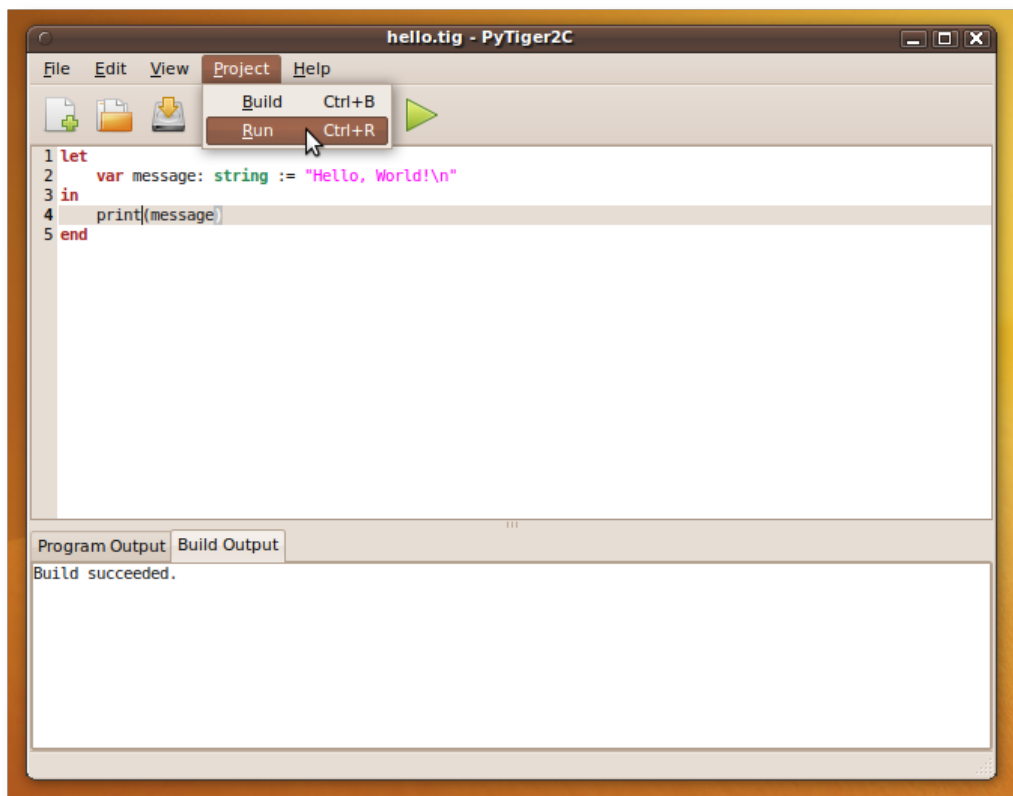


Figura 8: Ejecutando el nuevo programa.

La salida *standard* y de errores del programa se mostrará en la pestaña *Program Output* como se ilustra en la figura 9.

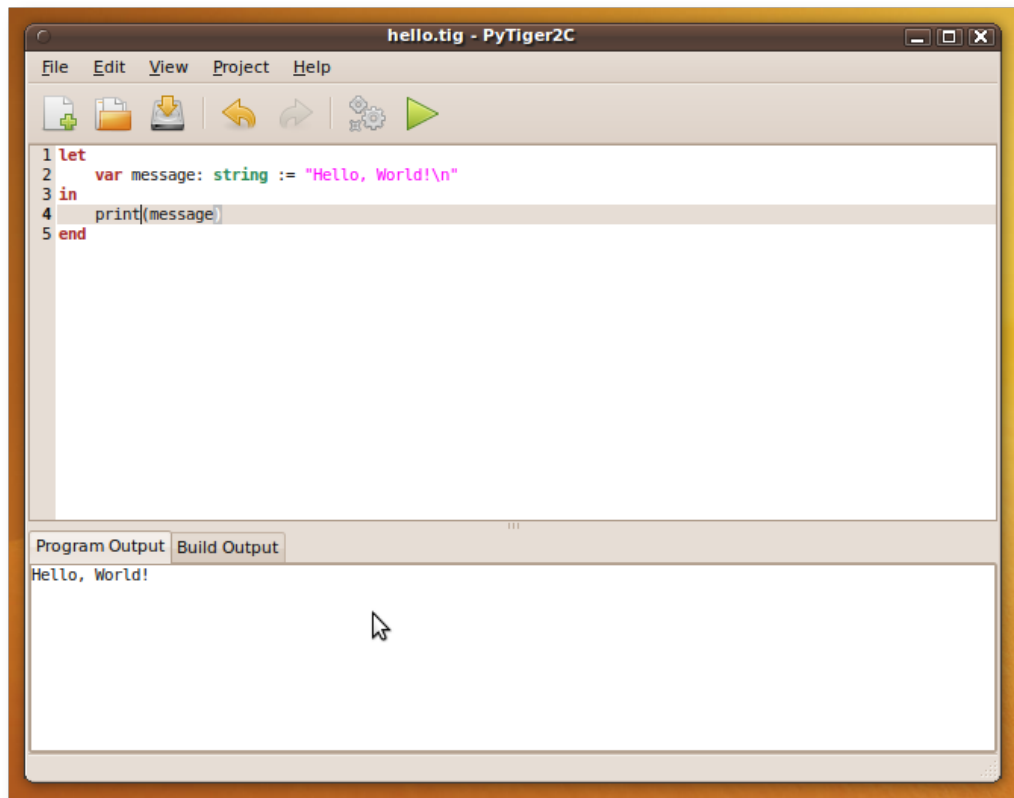


Figura 9: Salida del programa.

Es posible ver el código C generado para el programa Tiger durante el proceso de compilación. Para esto se utiliza el elemento *C Code* del menú *View* como se ilustra en la figura 10. Al hacer esto, se mostrará una ventana auxiliar con el código C como ilustra en la figura 11.

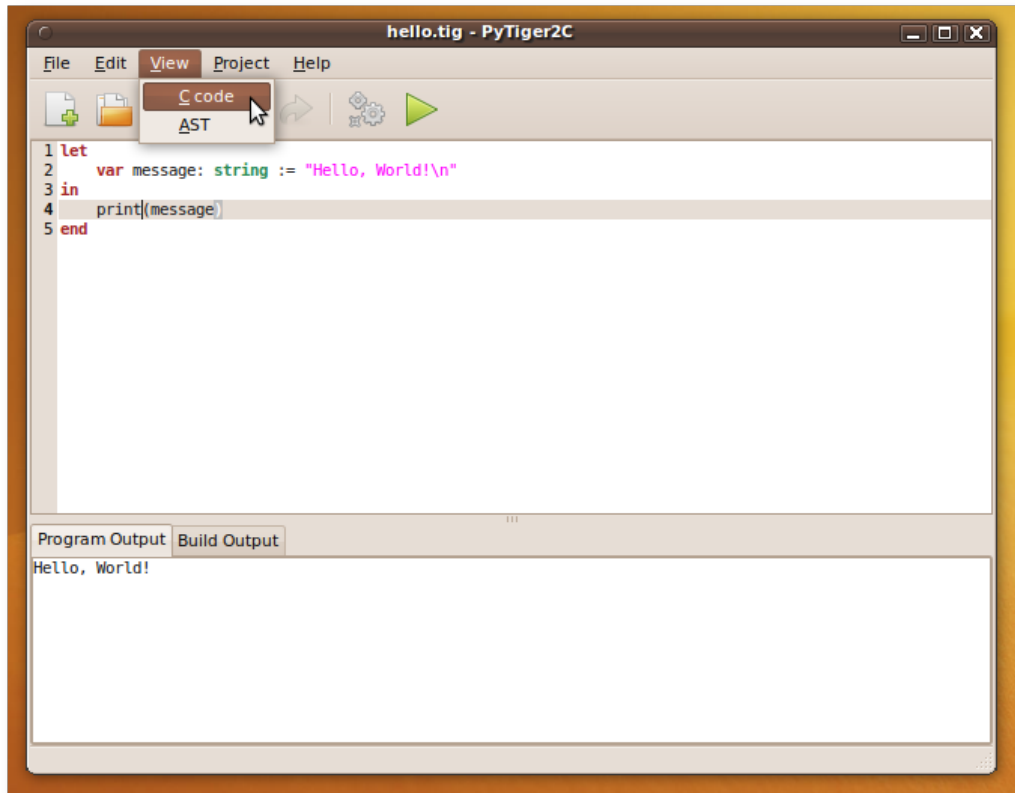


Figura 10: Ver el código C generado para el programa.

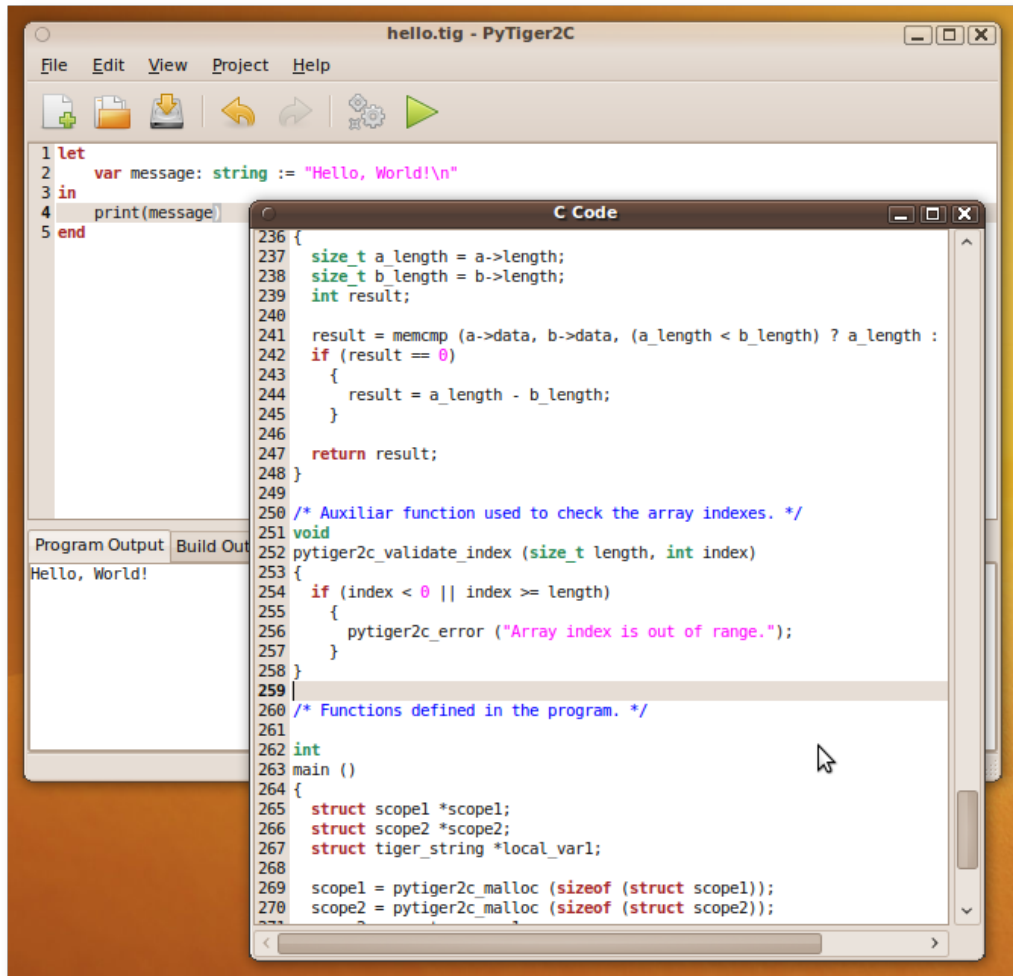


Figura 11: Ventana auxiliar mostrando el código C generado para el programa.

La interfaz gráfica permite además ver el árbol de sintáxis abstracta correspondiente al programa en edición. Esto puede hacerse mediante el elemento *AST* del menú *View* como se ilustra en la figura 12. Al hacer esto, se mostrará una ventana auxiliar con el árbol de sintáxis abstracta como se ilustra en la figura 13.

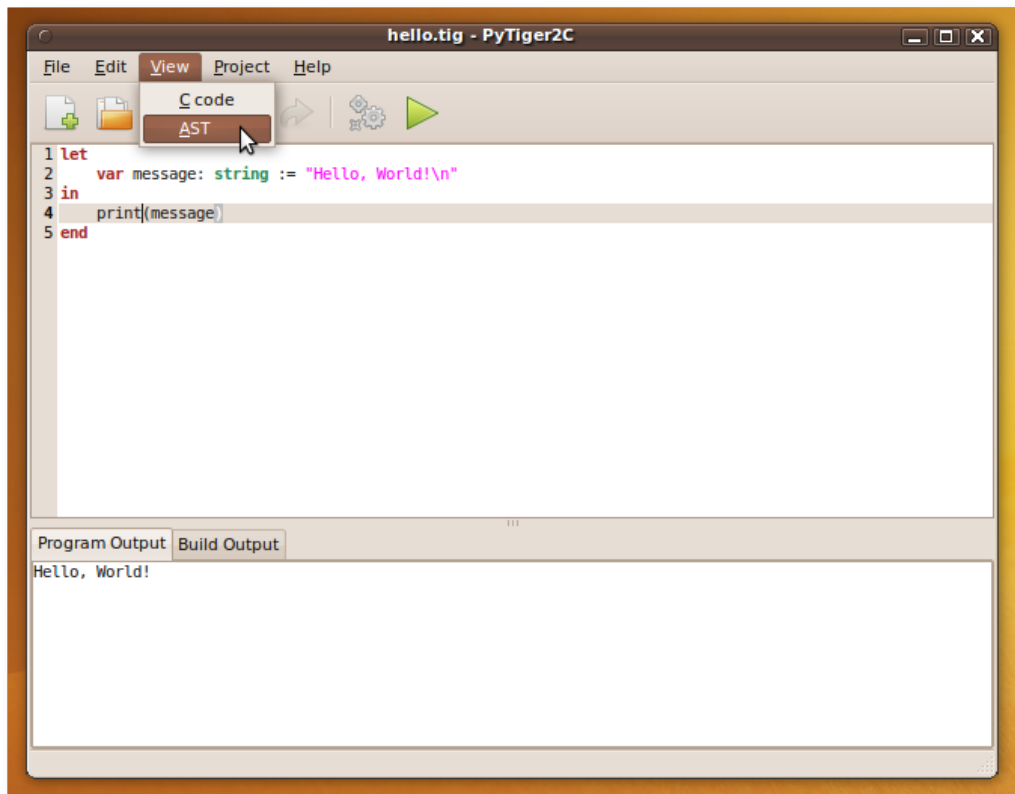


Figura 12: Ver el árbol de sintáxis abstracta correspondiente al programa.

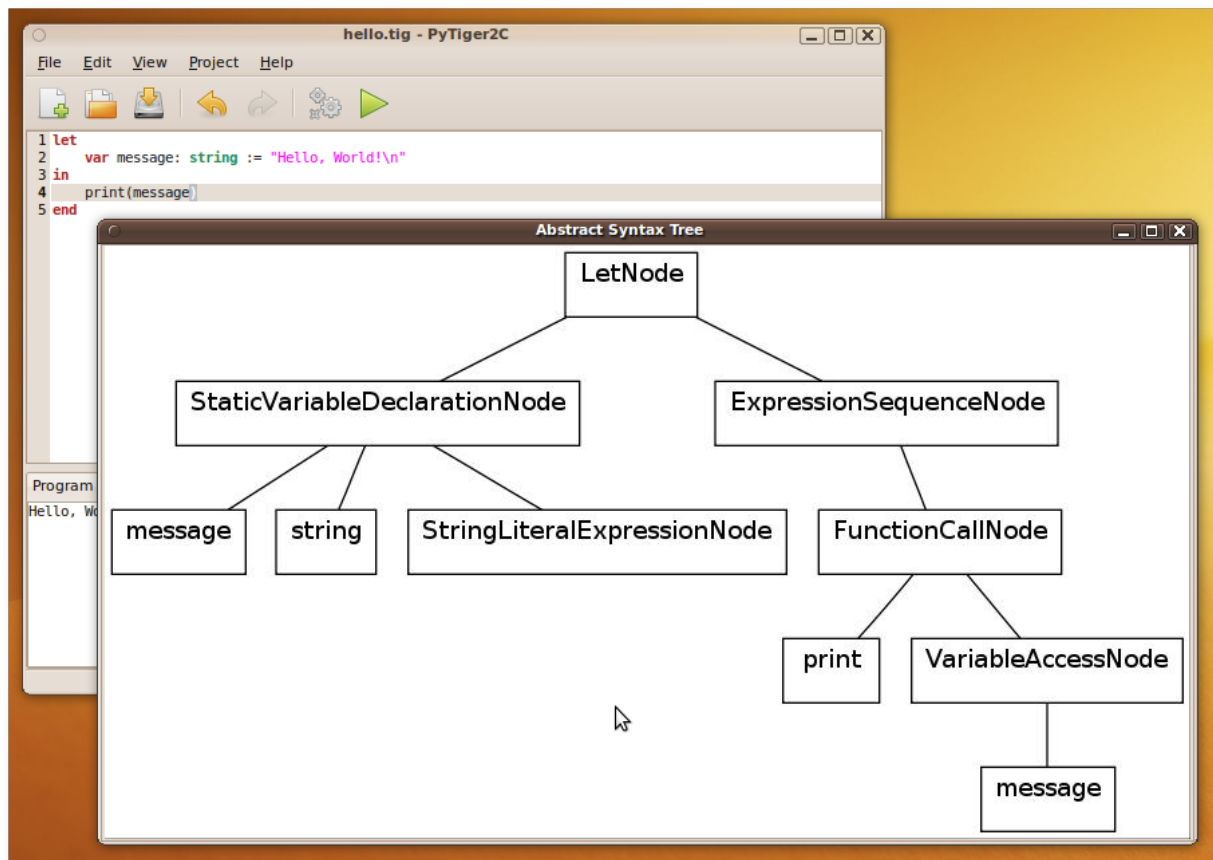


Figura 13: Ventana auxiliar mostrando el árbol de sintaxis abstracta correspondiente al programa.