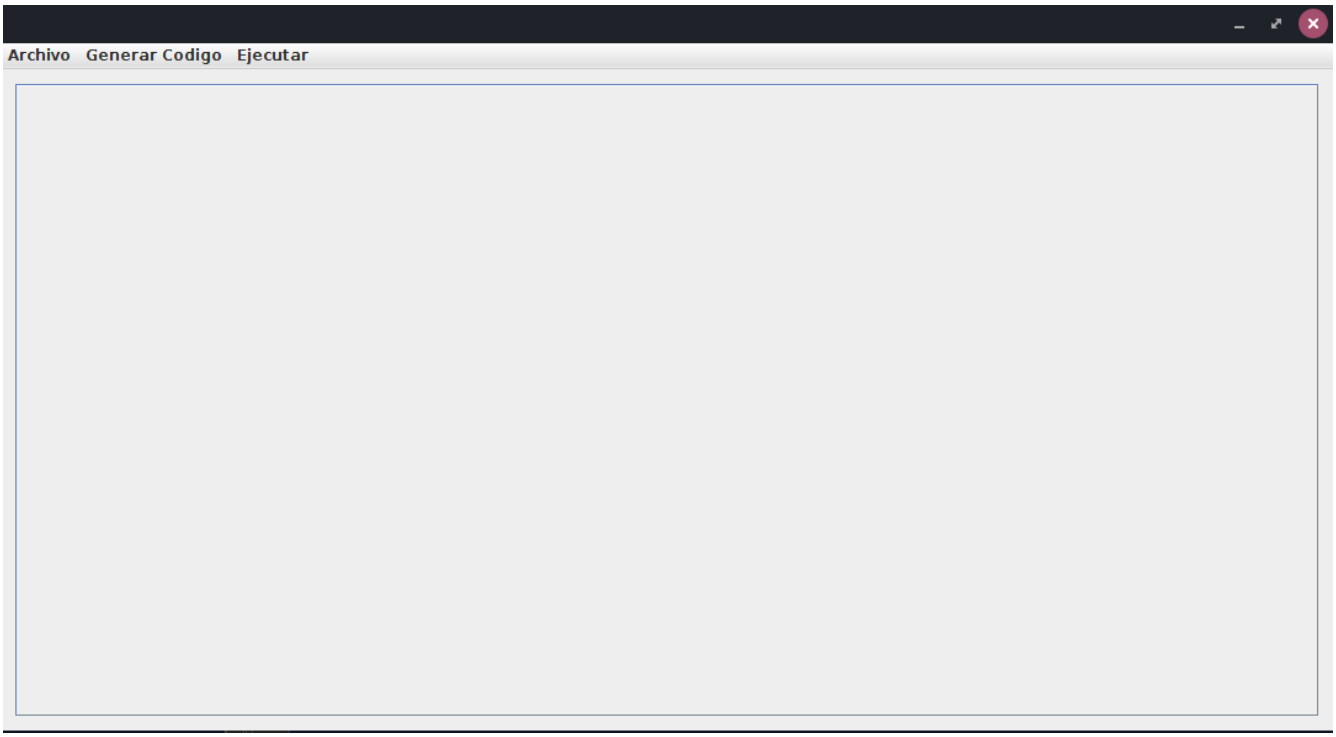
MANUAL DE USUARIO

La aplicación consiste en un traductor multilenguaje capaz de generar código intermedio en 3 direcciones, código 3 direcciones optimizado y código assembler.

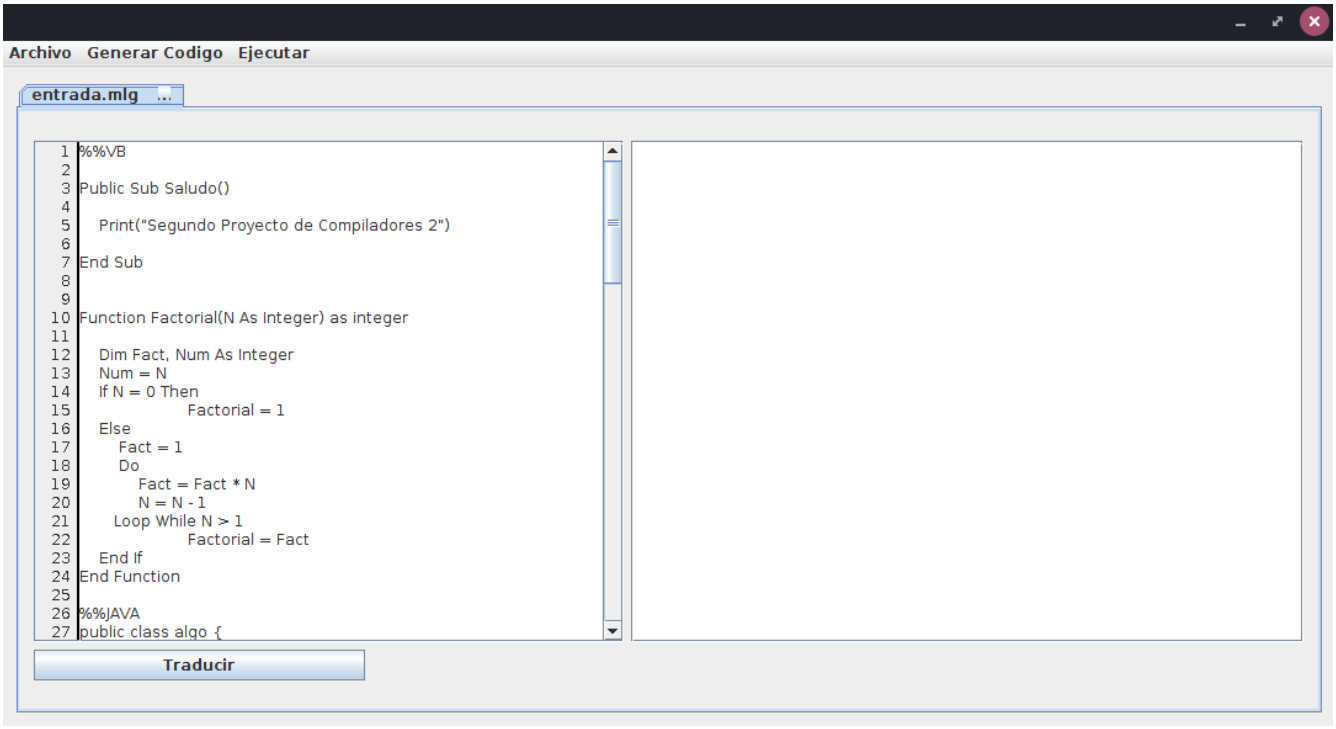


(Interfaz principal de la aplicación)

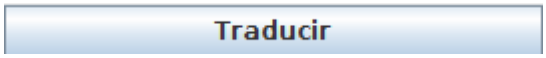
Una vez dentro de la aplicación se presenta una lista de opciones las cuales son:

ARCHIVO:

- **Nuevo:** Despliega la ventana denominada Programa, que contiene exclusivamente el código fuente del programa a compilar.
- **Abrir:** Esta opción permite abrir un archivo ‘.mlg’, este contiene el código fuente del programa almacenado con anterioridad.
- **Guardar:** Guarda el contenido de la ventana activa en la dirección que el usuario desee.
- **Guardar como:** Permite guardar el contenido de la ventana activa, con diferente nombre pero con la extensión que corresponde (,mlg).
- **Salir:** Finaliza la ejecución de la aplicación.



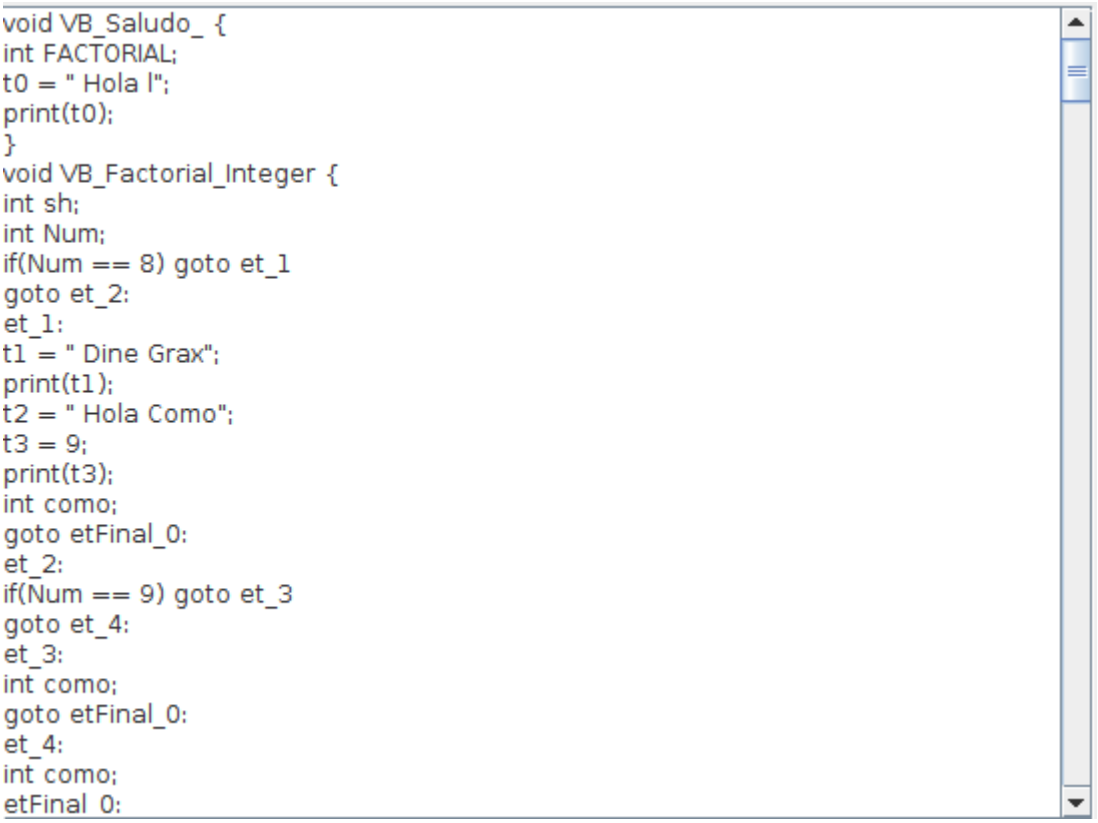
Para este proyecto una vez se abre un archivo .mlg se tienen dos ventanas paralelas una a la otra. La ventana del lado izquierdo corresponde a la ventana que contiene el código fuente abierto anteriormente, la ventana del lado derecho contendrá el código 3 direcciones al que se pretende llegar.



Traducir:
En dicho botón es donde se maneja todas las validaciones para llegar a la traducción a código 3 direcciones, SI Y SOLO SI el archivo de entrada no cuenta con ningún error de tipo: LÉXICO, SINTÁCTICO, SEMÁNTICO, se podrá traducir, de lo contrario cada error que presente se mostrado en la sección de Errores que viene dentro de la aplicación.

Sección de Errores:
Sección donde se muestra una descripción detallada de cada error que se llegue a encontrar dentro del archivo de entrada. El formato para mostrar los errores es:

Fila – columna – tipo de error – quien lo produjo – breve descripción



```
void VB_Saludo_ {
int FACTORIAL;
t0 = " Hola l";
print(t0);
}
void VB_Factorial_Integer {
int sh;
int Num;
if(Num == 8) goto et_1
goto et_2:
et_1:
t1 = " Dine Grax";
print(t1);
t2 = " Hola Como";
t3 = 9;
print(t3);
int como;
goto etFinal_0:
et_2:
if(Num == 9) goto et_3
goto et_4:
et_3:
int como;
goto etFinal_0:
et_4:
int como;
etFinal_0:
```

(Código 3 direcciones generado por el traductor)

ESTRUCTURA ARCHIVO .MLG

%%VB
    Modulos en Visual Basic
%%JAVA
    Clases en Java
%%PY
    Funciones y procedimientos en Python
%%PROGRAMA
    Sección de librerías de C
    Sección declaración de constantes
    Sección de Variables globales

```
void main()
{
    //Programa principal basado en lenguaje C
}
```

Dentro de la aplicación los lenguajes que conforman los modulos de la estructura principal son los siguientes:

- Visual Basic
- Java
- Python

Mientras que el programa principal utiliza C, para cada seccion del resto de la estructura se debe respetar la sintaxis de cada lenguaje dependiendo del modulo en el que se encuentre.