

# Augus

**Manual de Usuario**

Herbert Rafael Reyes Portillo  
201612114

# Indice

---

## Contenido

Indice .....	2
Introducción .....	3
Carga de Archivo .....	4
Reportes .....	6
<b>Reporte de Errores</b> .....	6
<b>Reporte de la Tabla de Símbolos</b> .....	7
<b>Reporte de AST</b> .....	8
Vistas Principales .....	10
<b>Especificaciones Generales</b> .....	10
<b>Consola</b> .....	13
<b>Debugger</b> .....	13
Flujo de Aplicación .....	15

# Introducción

---

Augus es un lenguaje de programación basado en PHP y en MIPS. Su principal funcionalidad es ser un lenguaje intermedio, ni de alto nivel como PHP ni de bajo nivel como el lenguaje ensamblador MIPS.

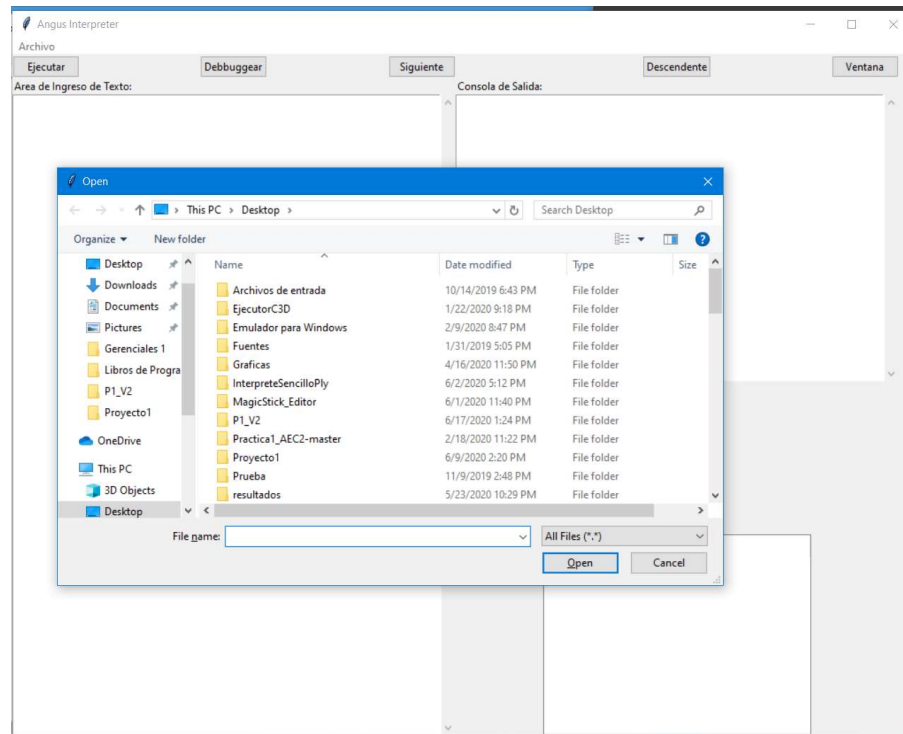
El lenguaje tiene dos restricciones: La primera es que cada instrucción es una operación simple, y la segunda, es que cada instrucción hay un máximo de dos operandos y su asignación.

Es un lenguaje débilmente tipado, sin embargo, si se reconocen cuatro tipos de datos no explícitos: entero, punto flotante, cadena de caracteres y arreglo.

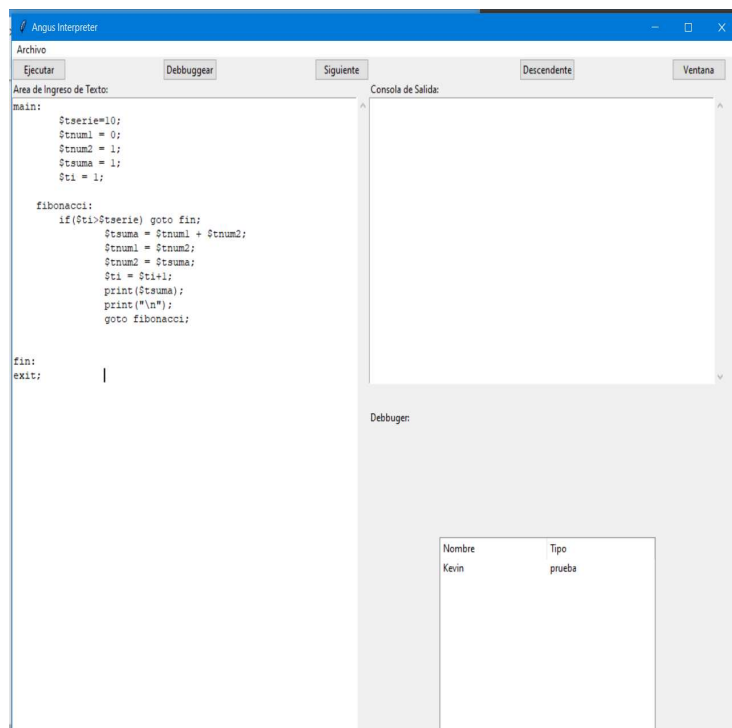
Para manejar el flujo de control se proporciona la declaración de etiquetas, sin tener palabras reservadas para ese uso.

# Carga de Archivo

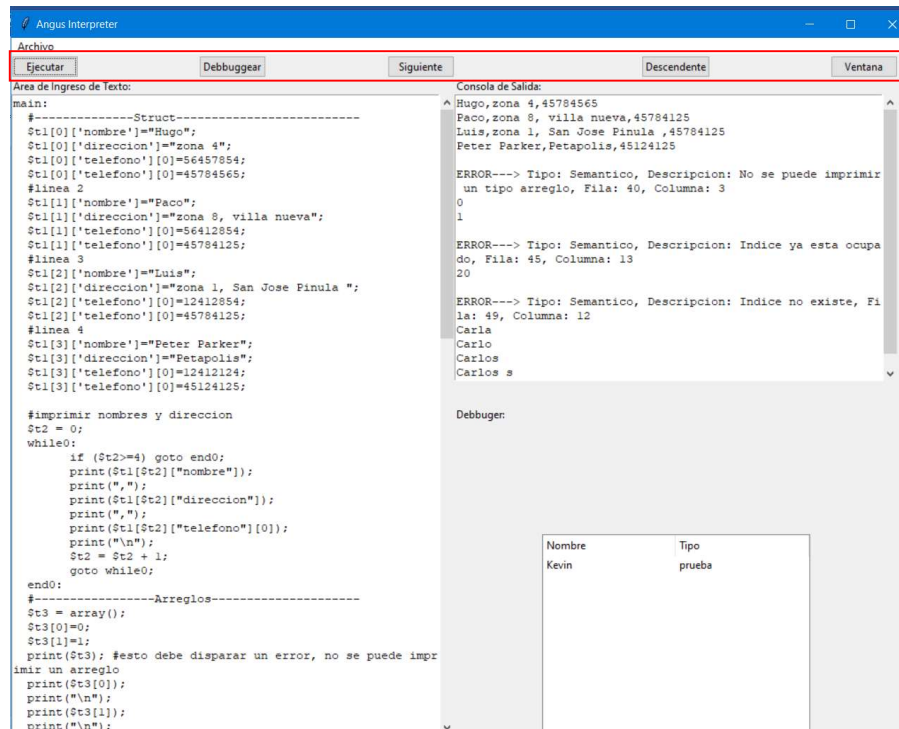
1. En el menú de archivo se encuentra la opción de abrir, se selecciona y esta despliega un cuadro de diálogo donde se debe seleccionar el archivo a analizar.



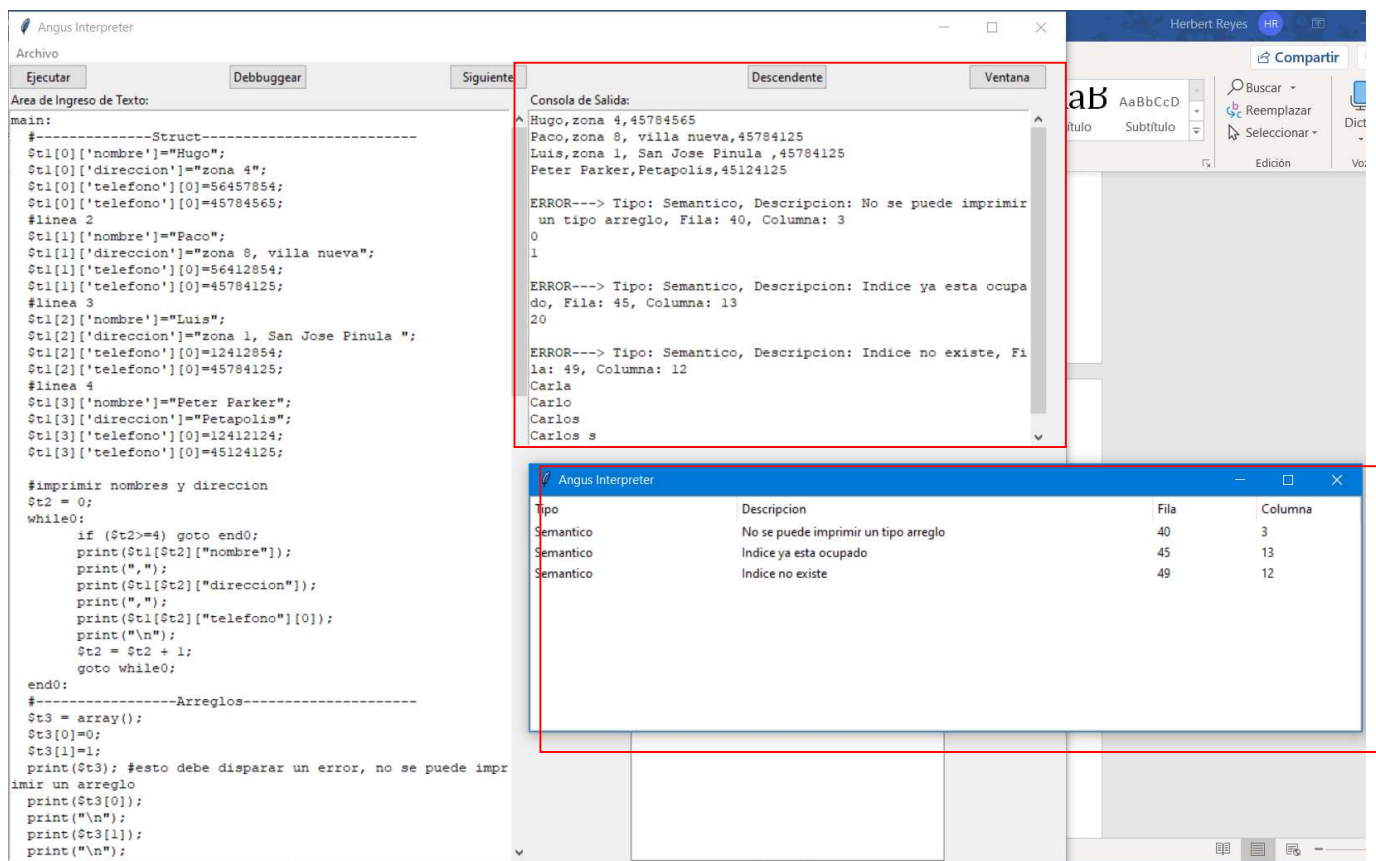
2. En la pestaña del lado izquierdo aparecerá el contenido del archivo seleccionado



- Para su análisis se debe seleccionar la forma que sea ascendente, descendente o debuggeado.



- A continuación, se puede ver la salida del texto analizado y los errores encontrados

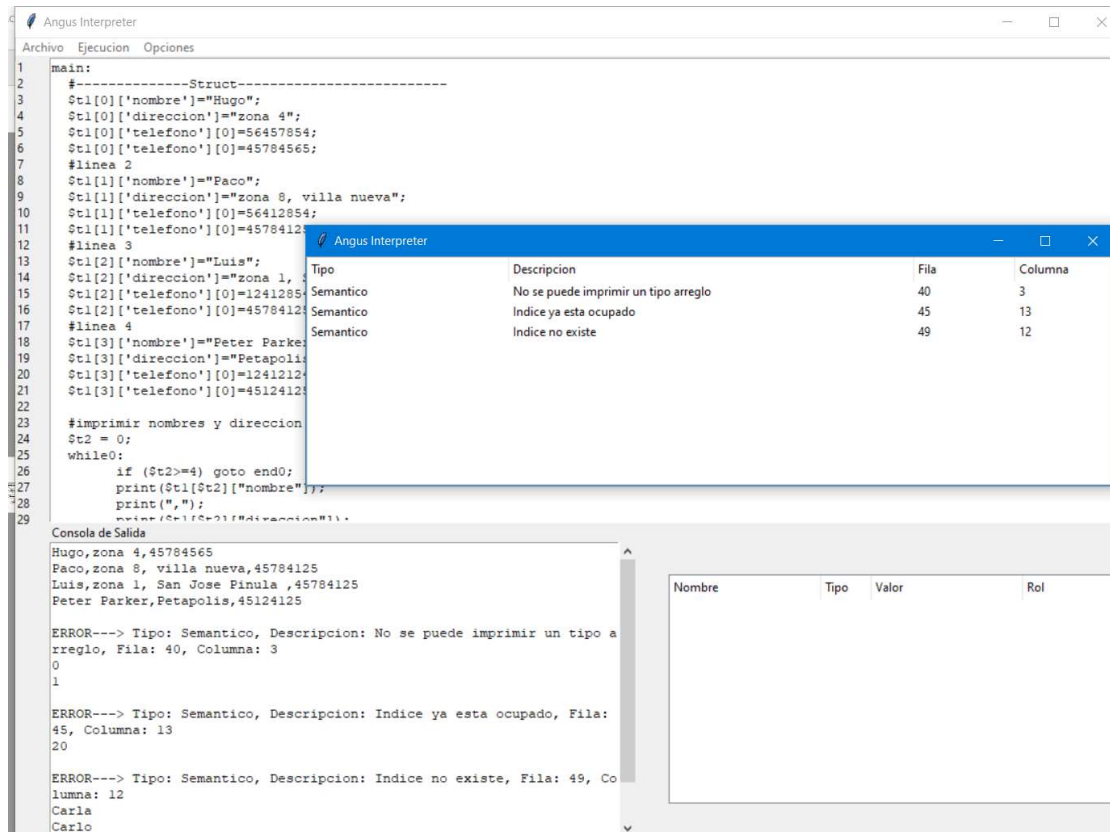


## Reportes

Después de Finalizada la Ejecución de los Reportes, automáticamente se le muestran al usuario los reportes generados.

### Reporte de Errores

El reporte de errores muestra los errores léxicos, sintácticos y semánticos encontrados al momento del análisis del archivo.



## Reporte de la Tabla de Símbolos

El reporte muestra todas las variables, funciones y procedimientos que fueron declarados, así como su tipo y toda la información relacionada a cada una de ellas.

Angus Interpreter

Archivo

Ejecucion

Opciones

```

1 main:
2 #-----Struct-----
3 $t1[0]['nombre']="Hugo";
4 $t1[0]['direccion']="zona 4";
5 $t1[0]['telefono'][0]=56457854;
6 $t1[0]['telefono'][0]=45784565;
7 #linea 2
8 $t1[1]['nombre']="Paco";
9 $t1[1]['direccion']="zona 8, villa nueva";
10 $t1[1]['telefono'][0]=56412854;
11 $t1[1]['telefono'][0]=45784125;
12 #linea 3
13 $t1[2]['nombre']="Luis";
14 $t1[2]['direccion']="zona 1, San J
15 $t1[2]['telefono'][0]=12412854;
16 $t1[2]['telefono'][0]=45784125;
17 #linea 4
18 $t1[3]['nombre']="Peter Parker";
19 $t1[3]['direccion']="Petapolis";
20 $t1[3]['telefono'][0]=12412124;
21 $t1[3]['telefono'][0]=45124125;
22
23 #imprimir nombres y direccion
24 $t2 = 0;
25 while 0:
26     if ($t2>=4) goto end0;
27     print($t1[$t2]["nombre"]);
28     print(",");
29     print($t1[$t2]["direccion"]);

```

Angus Interpreter

Nombre	Tipo	Valor	Declarada	Rol
\$t1	array	{0: {'nombre': 'Hugo', 'direccion': 'zona 4', 'telefono': {0: 56457854, 1: 45784565}}	main	temporal
\$t2	int	4	main	temporal
\$t3	array	{0: 0, 1: 1, 2: {0: 20}, 'nombre': 'Carlos'}	end0	temporal
\$s0	array	{0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0}	end0	pila
\$sp	int	-1	end0	puntero
\$a0	int	0	end0	parametro
\$a1	int	9	end0	parametro
\$t4	int	1	ack	temporal
\$v0	int	9	ret3	devfunc

Consola de Salida

```

Hugo,zona 4,45784565
Paco,zona 8, villa nueva,45784125
Luis,zona 1, San Jose Pinula ,45784125
Peter Parker,Petapolis,45124125

ERROR---> Tipo: Semantico, Descripcion: No se puede imprimir un tipo a
rreglo, Fila: 40, Columna: 3
0
1

ERROR---> Tipo: Semantico, Descripcion: Indice ya esta ocupado, Fila:
45, Columna: 13
20

ERROR---> Tipo: Semantico, Descripcion: Indice no existe, Fila: 49, Co
lumna: 12
Carla
Carlo

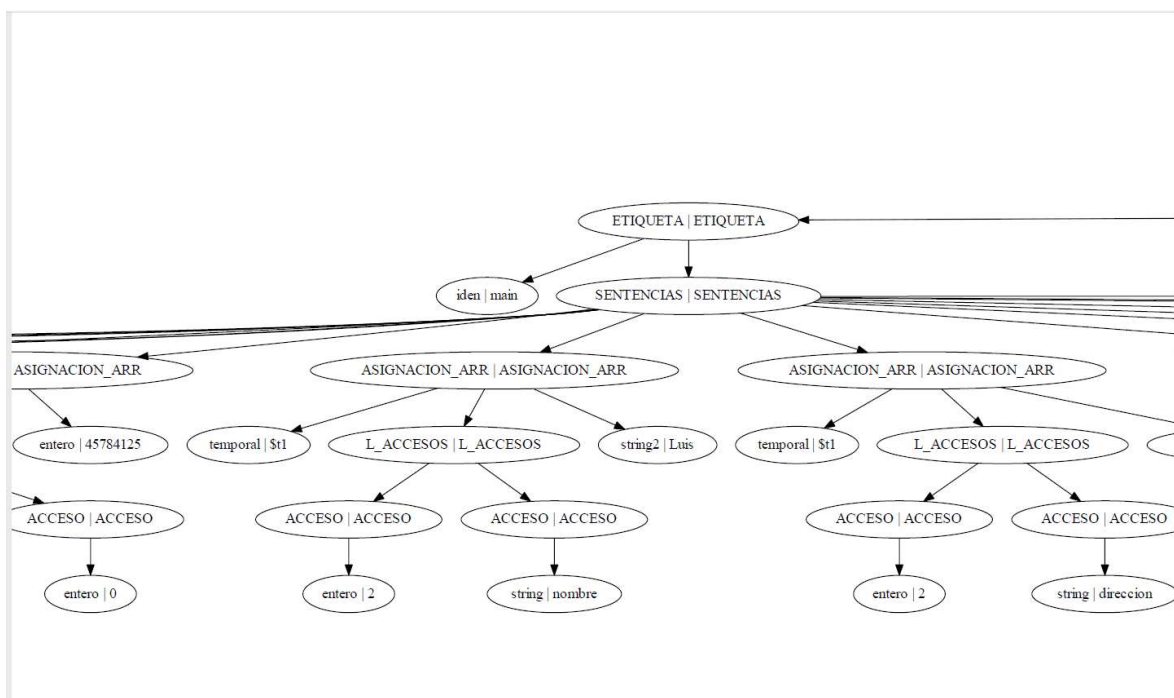
```

Nombre	Tipo	Valor	Rol
--------	------	-------	-----

## Reporte de AST

El reporte de AST muestra el árbol que se produjo en el análisis sintáctico:

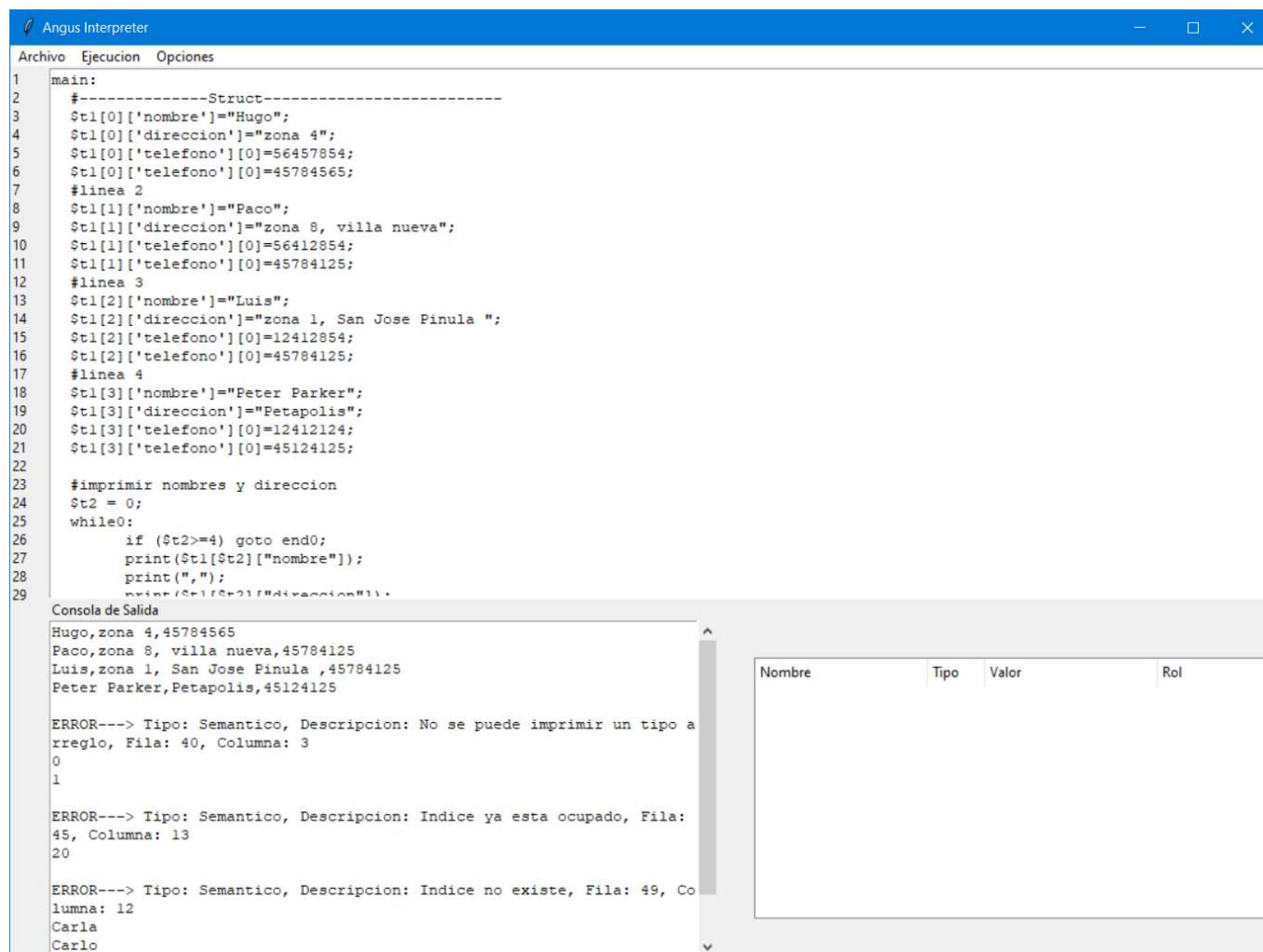




# Vistas Principales

## IDE AUGUS

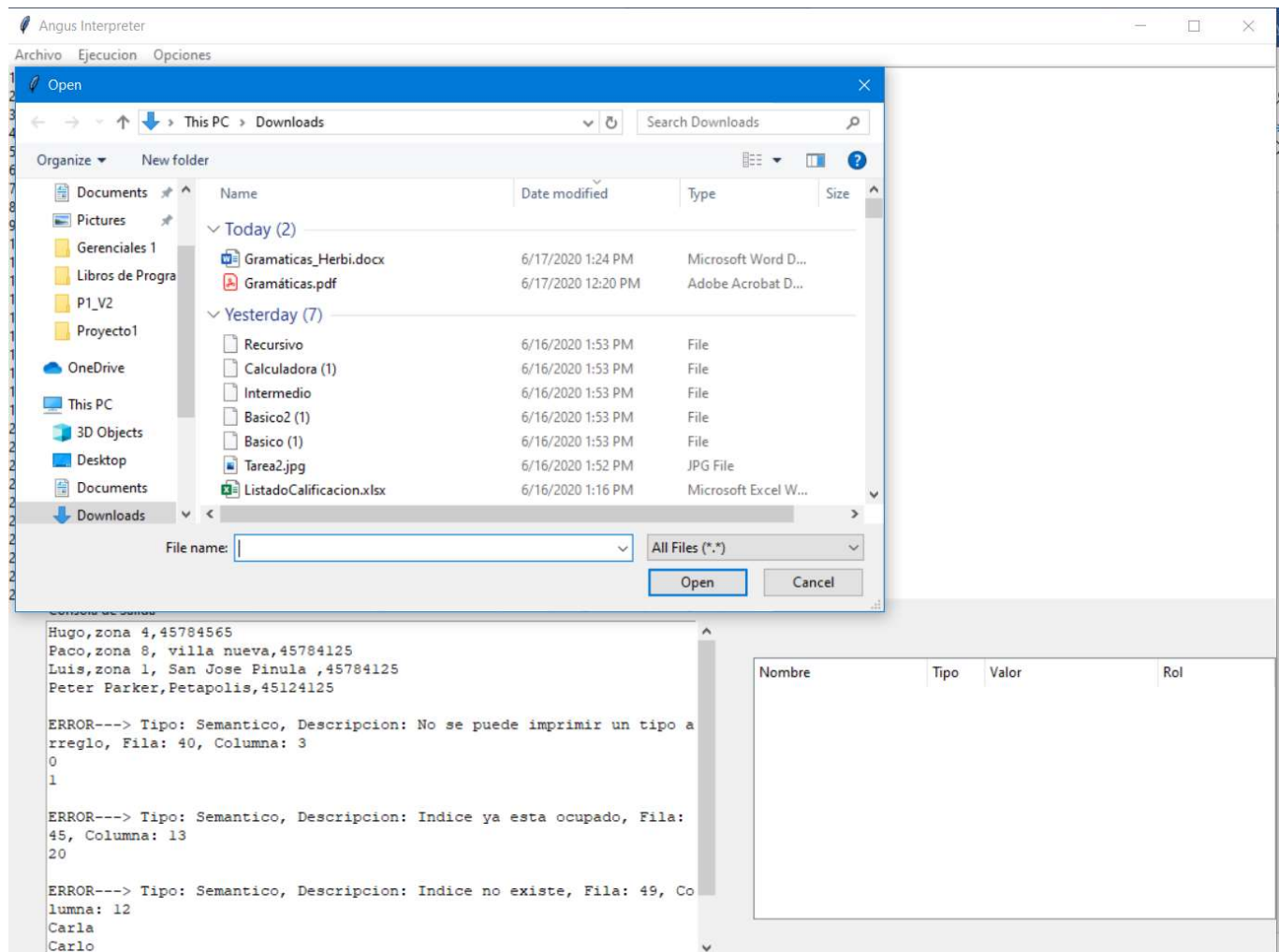
El IDE de Augus posee varias características para facilitar el análisis del código fuente, cada una de ellas será explicada a continuación.



## Especificaciones Generales

## Abrir:

Este botón abrirá un cuadro de dialogo para seleccionar un archivo existente y mostrará su contenido en la pestaña actual.

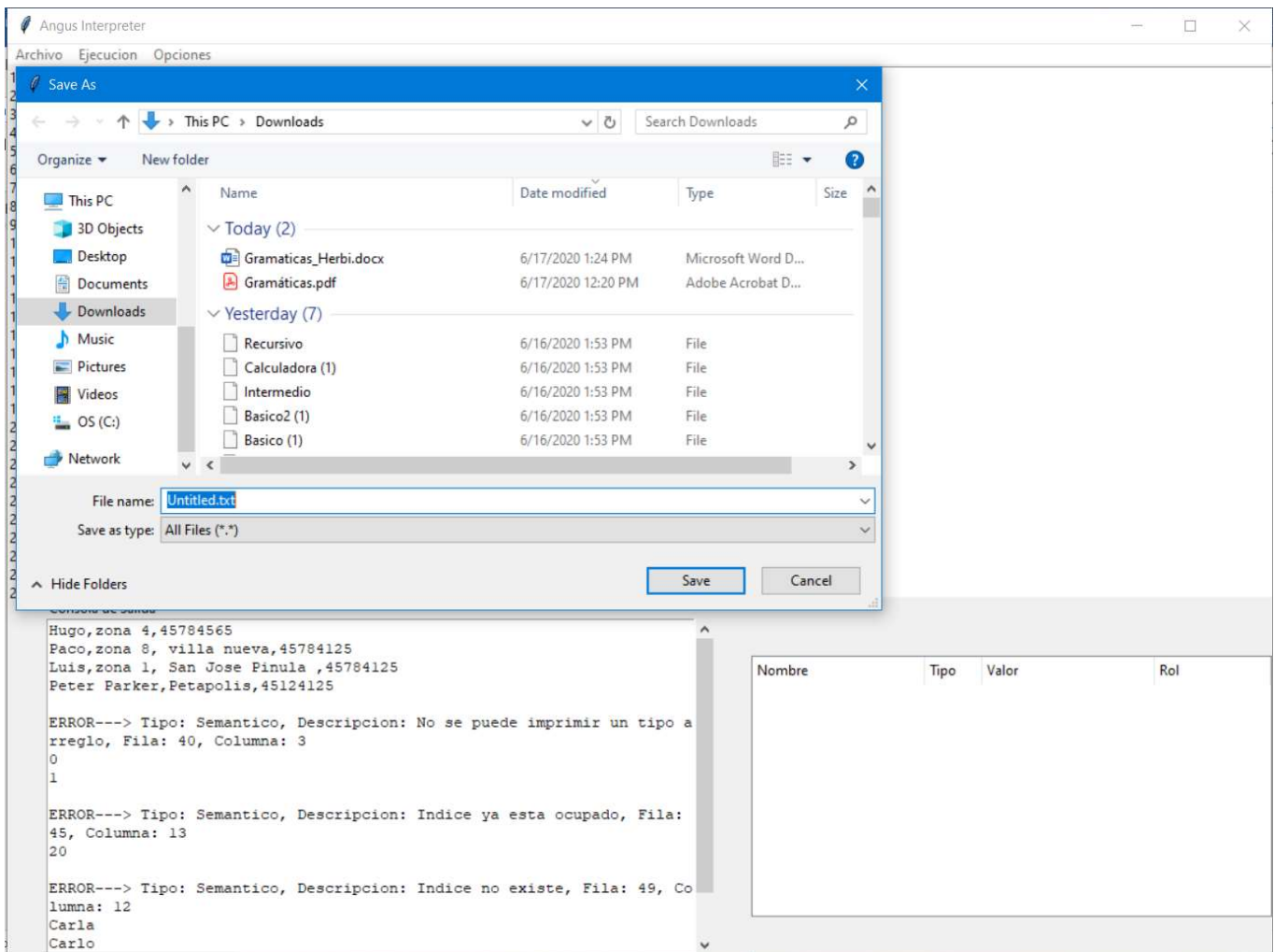


## Guardar:

Este botón permite guardar los cambios realizados en el archivo actual.

## Guardar Como:

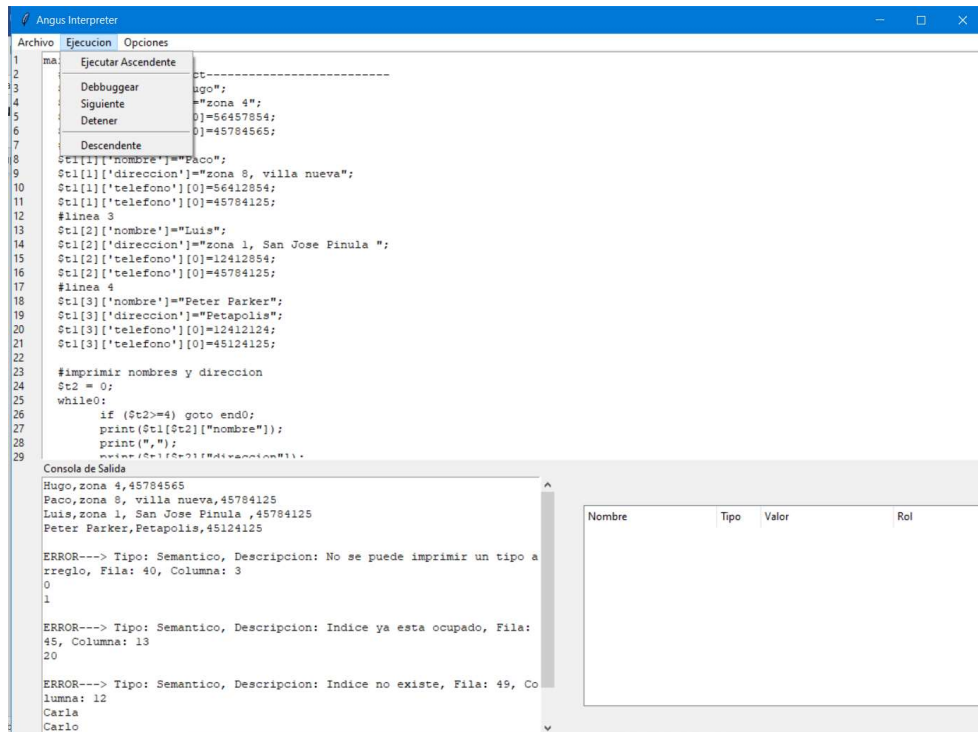
Al presionar este botón se mostrará un cuadro de dialogo para seleccionar la ubicación y el nombre del archivo a guardar.



### Botón de Ejecucion:

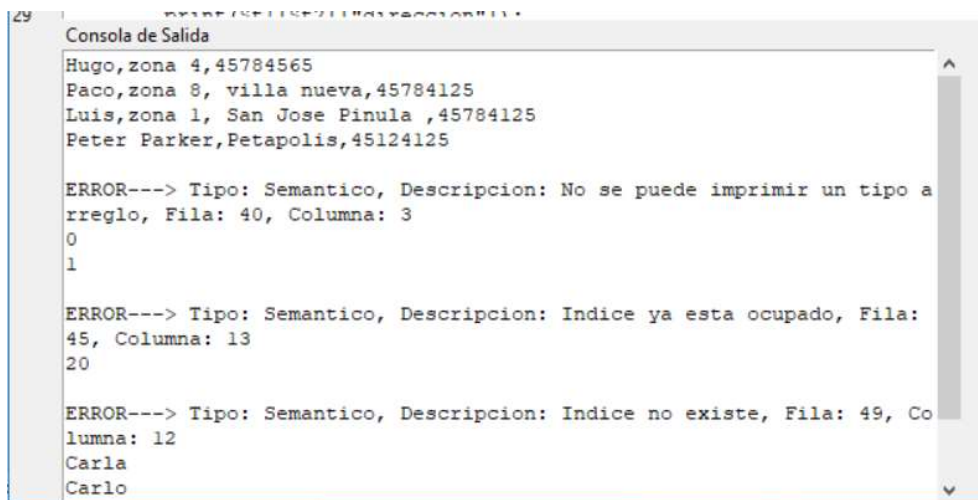
Arit permite la opción de elegir la forma en la que se analizará el archivo:

- Ascendente
- Descendente
- Debbuger



## Consola

Es el área donde se mostrarán los errores, advertencias o impresiones que se detectaron durante el transcurso del análisis de un archivo de entrada.



## Debugger

Cuando deseamos realizar el modo debugger basta con darle debugger en el menu.

Angus Interpreter

Archivo
Ejecucion
Opciones

```

1 main:
2 #-----Struct-----
3 $t1[0]['nombre']="Hugo";
4 $t1[0]['direccion']="zona 4";
5 $t1[0]['telefono'][0]=56457854;
6 $t1[0]['telefono'][0]=45784565;
7 #linea 2
8 $t1[1]['nombre']="Paco";
9 $t1[1]['direccion']="zona 8, villa nueva";
10 $t1[1]['telefono'][0]=56412854;
11 $t1[1]['telefono'][0]=45784125;
12 #linea 3
13 $t1[2]['nombre']="Luis";
14 $t1[2]['direccion']="zona 1, San Jose Pinula ";
15 $t1[2]['telefono'][0]=12412854;
16 $t1[2]['telefono'][0]=45784125;
17 #linea 4
18 $t1[3]['nombre']="Peter Parker";
19 $t1[3]['direccion']="Petapolis";
20 $t1[3]['telefono'][0]=12412124;
21 $t1[3]['telefono'][0]=45124125;
22
23 #imprimir nombres y direccion
24 $t2 = 0;
25 while0:
26     if ($t2>=4) goto end0;
27     print($t1[$t2]['nombre']);
28     print(",");
29     print($t1[$t2]['direccion']);

```

Consola de Salida

Nombre	Tipo	Valor	Rol
\$t1[0][nombre]	array	Hugo	temporal
\$t1[0][direccion]	array	zona 4	temporal
\$t1[0][telefono][0]	array	56457854	temporal
\$t1[0][telefono][0]	array	45784565	temporal
\$t1[1][nombre]	array	Paco	temporal

Nos ira indicando paso a paso que línea se va ejecutando y además, en la consola de debbuger tendremos como se muestra el avance de las variables en tiempo real, para ir continuando con el debbuger le damos siguiente y si deseamos pararlo, solamente basta con seleccionar la opción de detener.

## Flujo de Aplicación

---

Arit Software inicia su flujo de aplicación desde la carga del archivo que contiene el programa fuente hasta la creación de reportes o gráficas según lo analizado.

