

UNIVERSIDAD SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

Organización Lenguajes y Compiladores 1

Escuela de Ciencias y Sistemas

Segundo Semestre 2018

Ing. Mario Bautista, Manuel Castillo, César Batz

Tutores Académicos

Luis Paz, José Soc, Jorge Castañeda



PRÁCTICA 2: LENGUAJE FI

Contenido

PRÁCTICA 2: Lenguaje FI	1
OBJETIVOS	2
Objetivo General.....	2
Objetivo Específico:	2
DESCRIPCIÓN DE LA PRÁCTICA.....	2
INTERFAZ DEL LENGUAJE FI.....	3
Descripción de la interfaz:	3
DESCRIPCIÓN DEL LENGUAJE FI	4
1. Identificadores y variables:.....	4
2. Declaración de variables:	5
3. Tipos de operadores:	5
4. Métodos y llamadas:	6
5. Sentencias:	6
6. Manejo de tipos:.....	9
CONSIDERACIONES	10
1. Herramientas a utilizar:	10
2. Restricciones:	10
3. Entregables:.....	10
4. Fecha de Entrega:	10

OBJETIVOS

Objetivo General

- Que el estudiante comprenda el funcionamiento de un Árbol de Análisis Sintáctico y pueda usar este para implementar diferentes acciones de un lenguaje de programación estructurado.

Objetivo Específico:

- Que el estudiante sea capaz de poner en práctica los conocimientos adquiridos en el tema de Análisis Sintáctico mediante una aplicación.
- Que el estudiante sea capaz de construir gramáticas complejas y funcionales.
- Que el estudiante comprenda y utilice la herramienta Irony.

DESCRIPCIÓN DE LA PRÁCTICA

Como desarrollador, la empresa JAGU GT le solicita que desarrolle un lenguaje de programación estructurado el cual servirá como introducción para el desarrollo de aplicaciones más complejas que la empresa pretende implementar en un futuro, por lo que le solicita a usted que implemente la primera etapa del lenguaje, el cual se describe a continuación.

Es importante mencionar que la empresa le solicita el desarrollo de un lenguaje funcional el cual recibirá como nombre “**Lenguaje FI**” este debe ser capaz de realizar las acciones básicas descritas por los requerimientos de la empresa, las cuales incluyen declaración de variables, el manejo de ámbito de estas, uso de sentencias de flujo y control, uso de métodos y operaciones lógicas, relaciones y aritméticas.

INTERFAZ DEL LENGUAJE FI

Descripción de la interfaz:

La interfaz gráfica solicitada por la empresa se presenta a continuación:

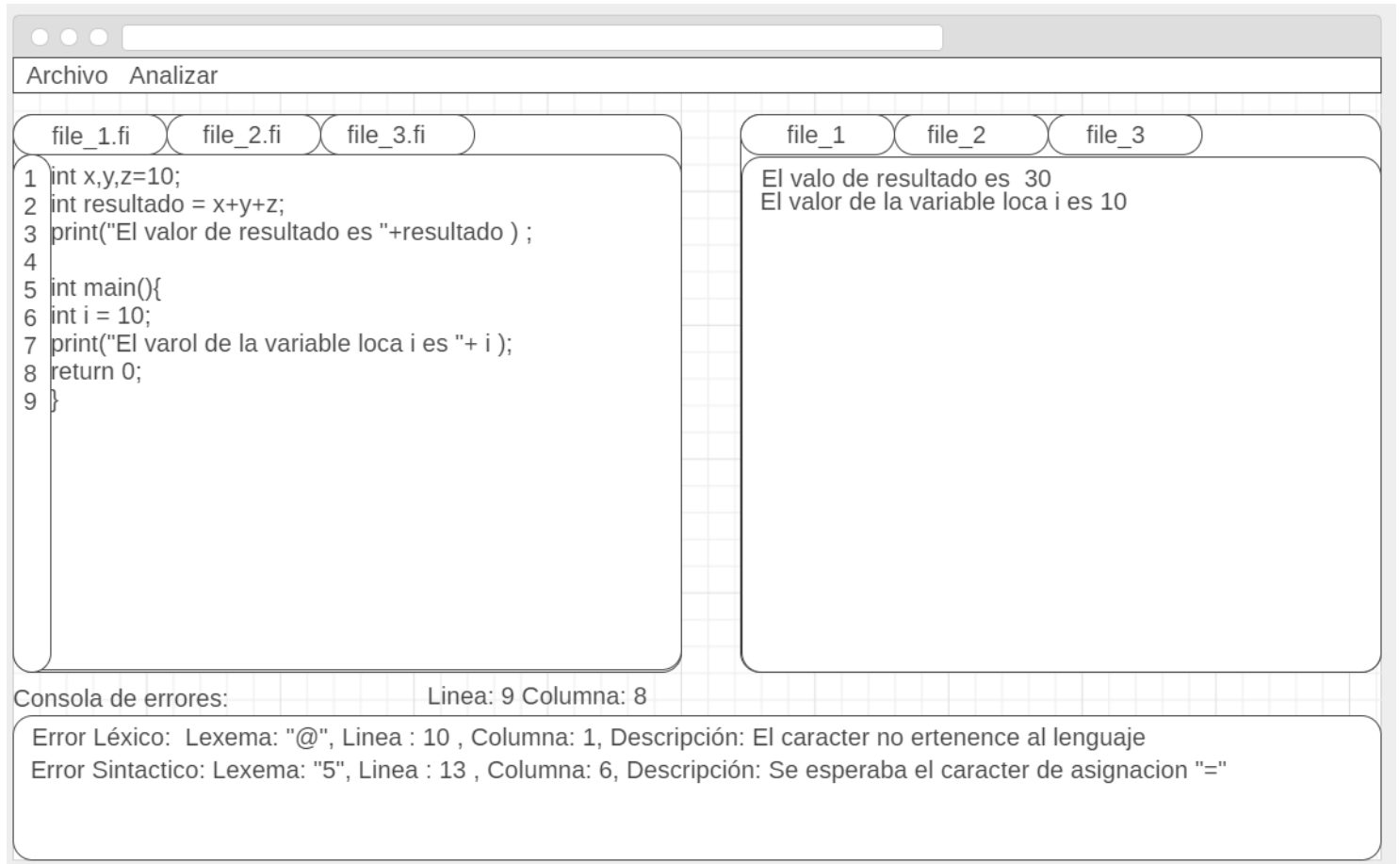


Imagen No. 1: Interfaz gráfica sugerida

Los botones (o menú) de la aplicación deberán contar con las siguientes funcionalidades:

1. **Archivo:** Creará una nueva pestaña en blanco en el editor.
 - 1.1 **Abrir:** Abrirá un cuadro de diálogo para seleccionar un archivo con extensión XML y mostrará su contenido en una nueva pestaña.
 - 1.2 **Cerrar pestaña:** Esta opción eliminará la pestaña de la interfaz gráfica.
2. **Analizar:** Analizará el contenido de la pestaña actual.
3. **Archivos Cargados:** En esta área se visualizarán los distintos archivos que han sido cargados a la aplicación, de forma que cada archivo está contenido en una pestaña diferente.

4. **Consola de salida:** En esta área se visualizarán los resultados de la función print que se encuentre en el archivo analizado, cada archivo tendrá su propia consola de salida por lo que también debe manejar múltiples pestañas.
5. **Consola de errores:** En este panel se podrán encontrar los errores encontrados en la fase de análisis léxico y sintáctico, por lo que se debe visualizar el número de fila y columna donde se encontró el error, el tipo de error, el lexema que lo causó y su descripción.

DESCRIPCIÓN DEL LENGUAJE FI

1. Identificadores y variables:

- **Identificadores:**

Consta de uno o más caracteres. El primer carácter debe ser una letra o el carácter subrayado (_), mientras que, todos los demás pueden ser letras, dígitos o el carácter subrayado (_). Las letras pueden ser minúsculas o mayúsculas.

- **Tipos de Dato**

Los tipos de datos son los siguientes

Tipo	Entrada
int	5,9,10,9999
float	5.5,10.00+52.333
char*	"cadena de caracteres"
bool	true,false

2. Declaración de variables:

Las variables serán declarables de la siguiente forma, las opciones dentro de corchetes son opcionales. Las variables se podrán declarar y acceder desde un ámbito global y local.

Forma 1:

<nombre_del_tipo_de_dato> <Lista de variables > [= <expresión>];

Forma 2:

<nombre_del_tipo_de_dato> <nombre_de_la_variable_1> [= <expresión_1>],

<nombre_de_la_variable_2> [= <expresión_2>],

...,

<nombre_de_la_variable_n> [= <expresión_n>];

Ejemplo:

```
int var1, var2;
```

```
int var1 = 10*10;
```

```
bool uno =true, dos= false, tres;
```

3. Tipos de operadores:

Las operaciones que se podrán realizar en el lenguaje son las siguientes:

- Operaciones Aritméticas:

Operador	Ejemplo
+	5 + var2
-	Var3-10
/	Var2/var4
*	45*10
%	50%2
- negativo	-5 + resta(10,2)
(agrupación)	((5+2)*(10-2)) +10

- Operaciones Relacionales:

Operador	Ejemplo
<	5.5 < 10*var2
>	50 > 60
<=	Var5 <= var5
>=	Var9 >= suma(5,2)
==	"cadena" == "cadena"

!=	"Pedro != "Luis"
----	------------------

- **Operaciones Lógicas:**

Operador	Ejemplo
(or)	true var2 5>10
&& (and)	true && (true false)

- **Operaciones especiales:**

Operador	Ejemplo
+=	i += 5*2;
-=	Var2-= 2+2;
--	j++;
++	int j = i++;

4. Métodos y llamadas:

Estos siempre retornan un valor, y la forma de ejecutarlos será invocándolos desde el método principal (main) a continuación se describe la sintaxis de estos.

```
<tipo> <nombre> ( [ <parámetros> ] ){
    < Lista Sentencias>
}
```

Ejemplo:

```
int suma(int num1, int num2){
    return num1+num2;
}
```

Llamada:

```
int res = suma(5,10*2) * 2;
```

5. Sentencias:

A continuación se describen las sentencias las cuales únicamente vendrán dentro de los métodos, los métodos también contendrán declaración y asignación de variables.

- **Return:**

Esta expresión es de tipo obligatorio por lo que todos los métodos deben retornar un valor del tipo del método.

```
return <expresión> ;
```

Ejemplo:

```
return num1+num2;
```

- **If-else:**

Esta instrucción se encargara de evaluar una condición, si esta se cumple procede a ejecutar su bloque de instrucciones sino se cumple y esta acompañada del bloque else, procederá a ejecutar las instrucciones este bloque.

```
if( <condicion> ) {  
    <instrucciones bloque if>  
} [ else {  
    <instrucciones bloque else>  
} ]
```

Ejemplo:

```
if(true || false){  
    print("Hola mundo");  
}  
  
if(false){  
    print("Entro al bloque if");  
}else{  
    print("entro al bloque else");  
}
```

- **While:**

Esta instrucción se encargara de realizar la ejecución de un bloque de instrucciones de forma cíclica mientras se cumpla la condición.

```
while(<condicion> ){  
    <instrucciones bloque while>  
}
```

Ejemplo:

```
while(i <10){  
    i++;  
}
```

- **Do-while:**

Esta instrucción se ejecutara al menos un vez, si la condición se cumple ejecutara las instrucciones de forma cíclica mientras se cumpla la condición.

```
do {  
    <instrucciones del bloque do>  
}while(<condicion>);
```

Ejemplo:

```
do{  
    print("Entro en un ciclo infinito");  
}while(true && true);
```

- **Comentarios:**

El lenguaje permitirá comentar el código por lo que para esto se tendrán dos tipos de comentarios, los cuales son comentarios de una sola línea o comentarios multilinea, estos podrán venir en cualquier parte del código.

Ejemplo:

```
while(i <10){  
    // este es un comentario simple  
    i++;  
    /* este es  
        un comntario  
        multilinea */  
}
```

- **Función print**

La función print es propia del lenguaje FI, por lo que por medio de esta función se podrá imprimir diferentes expresiones en la consola de salida del archivo analizado, por lo que se podrá hacer uso de la misma en cualquier bloque de código del lenguaje.


```
print ( <expresión> ) ;
```

Ejemplo:

```
print (var5);  
print("esta es una cadena simple");  
print( 5*5*10);  
print ("El valor resultantes es" + (5*5) );
```

6. Manejo de tipos:

- únicamente se podrán operar expresiones del mismo tipo.
- únicamente se podrán asignar expresiones a variables del mismo tipo.
- El operador + es el único que podrá ser utilizado como operador de concatenación entre cadenas de caracteres y otro tipo de expresión.
- Cualquier otra operación de tipos incompatibles debe ser reportada como un error.
- La ejecución dará inicio desde el método "main" y desde este se llamarán al resto de métodos o ejecutará las instrucciones contenidas en él.

CONSIDERACIONES

1. Herramientas a utilizar:

Para el desarrollo de esta práctica se utilizarán las siguientes herramientas.

- El lenguaje a utilizar es C#.
- El IDE a utilizar es Visual Studio (Windows) o su alterno (Linux).
- Para todo lo que concierne a análisis léxico y/o sintáctico se debe utilizar **irony**.

2. Restricciones:

Para el correcto desarrollo de esta práctica deben respetarse los siguientes lineamientos

- El análisis del archivo .FI es Case-Sensitive.
- Copias parciales o totales de proyectos tendrán una nota de 0 puntos y los estudiantes serán reportados a escuela de ciencias y sistemas.
- NO habrá prórroga de ningún tipo.
- La calificación se realizará a partir de los entregables enviados por el estudiante.

3. Entregables:

Los entregables para esta práctica son:

- Aplicación funcional
- Código fuente
- Gramáticas utilizadas en un archivo PDF que sea entendible.

4. Fecha de Entrega:

Entrega, 05 de octubre de 2018 antes de las 11:59 PM