



**COMPILADORES 1**

**PROYECTO FINAL**

**PRESENTADO A:**

ING. CARLOS VALLEJO

**ELABORADO POR:**

FRANKLIN GARCIA– 11611001

ALESSANDRO GRIMALDI – 11441226

**ASIGNATURA:**

COMPILADORES I

**LUGAR Y FECHA:** TEGUCIGALPA M.D.C. 29/09/2020

## **Introducción**

A través del presente documento presentamos el manual para el proyecto desarrollado en la clase de compiladores 1, teniendo en cuenta muchos aspectos de lenguajes de programación básicos como bucles, asignaciones y demás fueron implementados como se requiere según la clase, el proyecto se llama TATO y para este momento está desarrollado en las 2 primeras fases de un compilador las cuales son léxico y sintáctico

## **OBJETIVOS**

- Implementar los 2 primeros pasos de un compilador
- Crear un lexema para un lenguaje nuevo
- Implementar la parte sintactica del mismo lexema
- Conocer el concepto de un AST en implementarlo en el proyecto
- Implementar una serie de errores para que el programa detecte los errores posibles cuando se escribe y ejecuta un programa

**TABLA DE DESCRIPCION DEL LEXEMA IMPLEMENTADO Y EJEMPLOS SINTACTICOS  
PARA EL PROYECTO**

<i>Token</i>	<i>Función</i>	<i>Ejemplo</i>
Int	Integer	<b>int</b> id=4
Un dígito seguido de cero o mas dígitos	Números	1,2,3,432
Float	Float	<b>float</b> id=> 4.5
String	Cadena	<b>string</b> id=> "alo"
Bool	Boolean	<b>bool</b> id=> t
True	True	bool id=> <b>true</b>
False	False	bool id=> <b>false</b>
List	Arreglo de variables	<b>list</b> id(int)=>{1,2,3....., n}
Matriz	Arreglo bidimensional	<b>matriz</b> id(int)(int) =>{1,2,3..., n}{1,2,3..., n}
+, -	OPSUMA	id => 3 + 4
/	OPDIV	id => 3 / 4
*	OPMULT	id => 3 * 4
=>	OPASIGNACION	id => "a"
<,>=, >=,<=,!>=,!=,!	OPRELACION	a = b
If	Iniciador de bloque de decisión	<b>if</b>
end if	Fin del if	if condicion then .... <b>end if</b>
Else	Condición contraria a la dada	<b>Else</b>
end else	Fin del else	else then .... <b>end else</b>
else if	Siguiente condición	<b>else if</b> condición <b>then</b>
end else if	Fin del else if	else if condicion then .... <b>end if else</b>
Then	Make	if condición <b>then</b>
For	For	<b>for</b>
In	En	<b>in</b>
Range	Rango for	<b>range</b>
end for	Finalización del for	<b>end for</b>
While	Loop while	<b>while</b>
Repeat	Hacer loop while	<b>repeat</b>
end while	Fin del while	<b>end while</b>
;	PuntoComa	<b>;</b>
:	DosPuntos	<b>:</b>
,	Coma	<b>,</b>
(,)	Parentesis	<b>()</b>
" "	Comillas	<b>" "</b>
Return	Retorno	<b>return</b> 2;
Case	Switch	<b>case</b> (condicion) of
Of	Casos	case (condicion) <b>of</b>
Default	Default	Default
end case	Fin de caso	case (condicion) of .... <b>end case</b>

Main	Metodo main	<b>main</b> begin
Begin	Inicio funcion	main <b>begin</b>
end main	Fin main	main begin ... <b>end main</b>
Println	Imprimir con salto de linea	<b>Println</b> (2);
Print	Imprimir sin salto de linea	<b>Print</b> (2);
Call	Llamar metodo	<b>call</b> begin ImprimirHola ()
end call	Fin llamar metodo	call begin ImprimirHola() ..... <b>end call</b>
Una letra seguida de mas letras o numeros	Identificador	Int <b>a</b> =>3;
Palabras, numeros o caracteres dentro de Comillas	Cadena	String a=> <b>“hola-+=mundo123”</b>
And, Or	Operadores logicos	if (a>b)And (c>b)
Put	Funcion entrada	Put a;
/^	Comentario Entrada	<b>/^....^/</b>
^/	Comentario Salida	<b>/^....^/</b>
Ver coordenadas	VerCoordenadas de matriz o lista	Id[i][j] o id[i]

## ESTRUCTURAS IMPLEMENTADAS EN EL PROYECTO

### Bloque de decisión

```
if a=3 then
    Print(a);
end if
else if a<3 then
    a=>a+1;
    Print(a);
end else if
else then
    a=>a-1;
    Println(a);
end else
```

### Bloque de Iteración

#### **FOR:**

```
for Int i => 0 in range (i < a)
    Print(i);
end for
```

#### **WHILE:**

```
while true repeat
    Print("Hola mundo");
end while
```

## Switch/Case

**case(a) of**

**1:Println("hola");**

**2:Println("adios");**

**default:Println("nunca");**

**end case**

## Objetos

### Ejemplo (Main-Class-Methods):

**Class** = *program*

**Main** = *main begin*

*end main*

**Método** = *call*

*end call*

### METODO MAIN

main begin

Int a=>3;

if a=3 then

    for Int i=>0 in range (i<a)

        Print("hola");

    end for

end if

end main

### METODO VOID

call begin imprimirHola()

List lista(3)=>{1,2,3};

    for Int i=>0 in range(i<3)

        Print(lista[i]);

    end for

end call



## METODO CON RETORNO

```
Int begin multiplicarNumero(Int numero)
```

```
    Int multi=>numero*6;
```

```
    return multi;
```

```
end call
```

## Método Recursivo

```
Int begin metodoRecursivo(Int numero)
  Int resultado=>1+2*8;
  Int n=>1+2;
  numero=>resultado;
  if n=1 then
    return 1;
  end if
  else then
    numero=>numero-1;
    resultado => metodoRecursivo(numero);
    resultado=>resultado*numero;
    return resultado;
  end else
end call
```

## **Conclusiones**

- Se conocio y se implemento las primeras fases de un compilador lexico y sintactico
- Se implemento el AST
- Se creo un lenguaje basado en ideas de otros y nuevas