

**INFORME**

**TP FINAL COMPILADORES**

**ALUMNOS:**

* **Albornoz, Carlos 66**
* **Alegre, Gustavo 66**
* **Moura, Damian Elias 66422 d.eliasmoura@hotmail.com**

**PROFESOR:**

* **Ing. Ruidias, Hector**

**AÑO: 2018**

**INTRODUCCIÓN**

Mediante este informe se busca describir los pasos llevados a cabo para el desarrollo de un compilador (análisis léxico y sintáctico), que traduce un lenguaje fuente escrita en pseudocódigo a un lenguaje comprensible por la máquina. El lenguaje elegido por los desarrolladores al cual se traduciría el lenguaje fuente escrito en pseudocódigo es el Lenguaje C.

**LENGUAJE FUENTE**

Inicio Nomb\_Programa

Sección de declaración de constantes

Sección de declaración de variables

Sección de sentencias

Fin

La sección de declaración de constantes se utiliza para declarar todas las constantes numéricas o de carácter que se van a utilizar durante el programa.

La sección de declaración de variables se utiliza para declarar todos los identificadores que se van a utilizar durante el programa.

Cada declaración se hace de la siguiente manera:

Variables:

identificador1, identificador2,... tipo tipo1

identificador3, identificador4,... tipo tipo2

identificador\_vector[] tipo tipo3

1. Los identificadores pueden ser cadenas de caracteres alfabéticos o bien pueden comenzar por una letra, seguida de uno o más letras o dígitos.
2. Los tipos disponibles son:

- Numerico: un número, que puede ser entero o entero con dos decimales

- Cadena: un conjunto de caracteres.

La sección de sentencias permite la escritura de las instrucciones del programa, cada sentencia debe escribirse en una nueva línea y debe finalizar con una coma al final. Las sentencias pueden ser simples o compuestas. A saber:

1. Sentencia simple:
   1. Instrucción de asignación. Se la declara de la siguiente manera:
      1. identificador=expresion\_matematica,
      2. identificador=constante,
      3. identificador=identificador,
      4. identificador = [valor 1 valor 2 valor 3 ...valor n ],

importante: al asignar un arreglo se debe tener en cuenta que todos los valores sean del mismo tipo que el tipo del que fue declarado en la sección de declaración de variables

* + 1. identificador= (expresion\_logica)? expresion\_por\_si:expresion\_por\_no,

1. Instrucción de entrada/salida. Se la declara de la siguiente manera:
   1. Imprimir(cadena) imprime un mensaje por pantalla
   2. bitacora(cadena) imprime un mensaje del estado de la aplicación en el log del sistema o del runtime
   3. leer(variable) pide un valor al usuario y lo guarda en variable
   4. cargar\_vector(archivo, variable) lee desde un archivo con formato valor1, valor2, valor3,.....,valorn y lo almacena en variable como un vector
2. Sentencia compuesta: sentencias condicionales y las repetitivas. A saber:
   1. Condicionales:
      1. Si expresión\_lógica

sentencia1,

sentencia2,...

sino

sentencia1,

sentencia2,...

Finsi

* + 1. evaluar variable

caso valor1:

sentencia,

sentencia 2,...

caso valor2:

sentencia,

sentencia 2,...

otro

sentencia 1,

sentencia 2,...

finevaluar

* 1. Repetitivas:
     1. Mientras expresion\_lógica

sentencia1,

sentencia2,...

Finmientras

* + 1. interar arreglo:interador

sentencia 1,

sentencia 2,...

* + 1. finiterar

Una expresión se define de la siguiente manera:

* identificador operador identificador o
* identificador operador constante o
* constante operador identificador o
* constante operador constante

y puede ser de dos tipos:

- Matemática: cuando se realiza una operación matemática. Puede ejecutarse entre constantes y/o identificadores, siempre y cuando ambos sean numéricos.

Los operadores disponibles para declarar una expresión numérica son **+, -, \* y /.**

- Lógica: cuando se realiza una comparación. Puede ejecutarse entre constantes y/o identificadores de cualquier tipo.

Los operadores disponibles son **<, >, <=, >=, != e ==.**

Las constantes, al igual que los identificadores, pueden ser caracteres, numéricas o cadena.

También se puede incluir comentarios en el código fuente de la siguiente manera

**!! comentario 1**

este es un comentario de una sola línea

**!!%**

**comentario 1**

**%**