Compiladores

Práctica 1 - Analizador léxico

**Objetivo:**

Construir un analizador léxico utilizando funciones.

**Definiciones:**

Un analizador léxico lee los caracteres de la entrada y los agrupa en ***"objetos token"***.

**Ejercicios**

Tenemos un lenguaje que acepta las siguientes expresiones:

> 1+2

> 4\*3-8

> 60 / 5 + 3 - 0

> A=12

> variable1 =14

> C = A + variable1

1. Escriba una función que analice, caracter por caracter.

const char \*linea = “1+2”;

vector<Token> tokens = **analizadorLexico**( linea );

1. Escriba dos funciones que reconozcan un número y el nombre de una variable.
   1. Un número es solo un número entero
   2. El número entero solo tiene dígitos [0-9]
   3. El nombre de una variable puede ser:

[a-z | A-Z][0-9 | a-z | A-Z]\*

1. Las funciones son:

Token **reconoceNumero**(char \*\*ptr);

Token **reconoceVariable**(char \*\*ptr);

**Orientación inicial:**

vector<Token> **analizadorLexico**( char \*linea ) {

vector<Token> tokens;

switch( \*linea ) {

case ‘0’: case ‘1’: …. case ‘9’:

Token token = reconoceNumero( &linea)

tokens.push\_back(token);

break;

…

**default: linea++; // Significa que encontró un espacio en blanco**

}

return tokens;

}

class Token {

**char** \*palabra; //almacena una copia de la palabra

**int** indice;

**char** tipo; //E (entero), V (variable), O (operador)

};