Análisis Léxico

El análisis léxico tiene como propósito transformar una secuencia de caracteres del programa fuente en una secuencia de *tokens*.
Un *token* es una unidad con significado dentro del lenguaje de programación, y está compuesto por un *lexema*, que es la secuencia concreta de caracteres que lo conforma.

El proceso lo realiza el *lexer* o analizador léxico, que toma la entrada (código fuente) y produce como salida los *tokens* que servirán de entrada al *parser* (analizador sintáctico).
En teoría, no sería indispensable tener un *lexer*, ya que todo lo que este hace también puede hacerlo el *parser*. Sin embargo, se suele utilizar un *lexer* porque simplifica el diseño, reduce la complejidad y resulta más eficiente desde el punto de vista de la ingeniería de software.

Eliminación de espacios en blanco y comentarios

Los espacios en blanco y los comentarios no se convierten en *tokens*, por lo que el *parser* no necesita preocuparse por ellos. El *lexer* se encarga de descartarlos durante la lectura del código fuente.

Lectura anticipada

En algunos casos, es necesario leer más de un carácter para determinar el *token* correcto.
Por ejemplo, se debe distinguir entre un operador simple como `<` y uno compuesto como `<=`.
En esta situación, después de leer un carácter, puede ser necesario examinar el siguiente. Si el carácter adicional no forma parte del *token*, debe devolverse a la entrada para que pueda ser procesado más adelante.
Este mecanismo se conoce como *lectura anticipada*.

Constantes

Considerando únicamente las constantes numéricas enteras, su valor se construye dígito por dígito con la regla:

```
valor := 10 * valor + dígito
```

De esta forma, el *parser* recibe el *token* `num` en lugar de una simple secuencia de dígitos.
El valor de la constante se maneja como atributo del *token* `num`. Este atributo puede ser directamente el valor numérico o bien una referencia a la entrada correspondiente en una tabla de símbolos que almacene los números.

Reconocimiento de identificadores y palabras clave

El analizador léxico debe identificar y diferenciar los identificadores de las palabras clave.
Aunque ambos tienen la misma forma sintáctica (una cadena de caracteres), los identificadores representan variables o nombres definidos por el usuario, mientras que las palabras clave forman parte del lenguaje y tienen un significado especial.

Para resolver esta distinción se utiliza la tabla de símbolos.
Las palabras clave se cargan previamente en esta tabla y se marcan como tales. Así, cuando el *lexer* encuentra una cadena que parece un identificador, consulta la tabla: si está registrada como palabra clave, se clasifica de esa manera; si no, se reconoce como identificador.

El *lexer* también debe manejar situaciones en las que un *lexema* sea un subconjunto de otro, como en la diferencia entre `<` y `<=`, o entre `then` (palabra clave) y `thenewvalue` (identificador).

Un analizador léxico

El *scanner* o analizador léxico tiene la función de reconocer *tokens* como números, identificadores, palabras clave y operadores, transformando la secuencia de caracteres en una representación más estructurada para el *parser*.

Al reconocer números, se genera el *token* `num`, cuyo atributo `value` contiene el valor numérico correspondiente. Esto permite que la gramática se simplifique, ya que el *parser* trabaja con *tokens*

en lugar de caracteres individuales.

La gramática puede representarse de forma más compacta al introducir niveles como `factor`, lo cual permite manejar expresiones con operadores de distinta precedencia e incluir el uso de paréntesis.
El procedimiento de análisis sigue el esquema de descenso recursivo: se selecciona la producción que corresponde según el símbolo de entrada, se invocan los procedimientos adecuados para los no terminales, se comparan los terminales esperados y se ejecutan las acciones semánticas asociadas.

De esta manera, el analizador léxico permite procesar constantes de más de un dígito, identificadores, palabras clave y operadores, facilitando la tarea del analizador sintáctico.