



UNIVERSIDAD DE  
GUADALAJARA



## Actividad #1 ¿Qué es un analizador léxico?

**Alumnos:** Zashuvath López Moreno Ethan Israel

**Código:** 216493953

**Materia:** Seminario de Solución de Problemas de Traductores de Lenguaje 2

**Sección:** D02

**Aula:** DUCT2-7

**Docente:** López Franco Michel Emanuel

## Contenido

|   |   |
|---|---|
| Introducción:.....                                  | 3 |
| Funciones Principales:.....                         | 3 |
| Entrada para el Analizador Sintáctico:.....         | 3 |
| Reconocimiento basado en Gramáticas Regulares:..... | 3 |
| Generación de Componentes Léxicos:.....             | 3 |
| Primera Fase del Compilador:.....                   | 4 |
| Conclusión: .....                                   | 4 |

## Introducción:

El análisis léxico, también conocido como lexing o tokenización, es como el primer paso al compilar un programa. En pocas palabras, convierte una secuencia de letras y símbolos en unidades más manejables llamadas tokens. Esto lo hace un analizador léxico, también conocido como lexer. En este informe, vamos a hablar sobre por qué esto es tan importante en el desarrollo de programas.

## Funciones Principales:

Imagina que el código fuente de un programa es como un montón de letras y números complicados. El trabajo del analizador léxico es desglosar todo eso en pedacitos más simples llamados tokens. Cada token representa algo específico en el código, como un nombre, un operador o una palabra clave. Esto facilita mucho las siguientes etapas del proceso de compilación.

## Entrada para el Analizador Sintáctico:

Lo que saca el analizador léxico, que son estos tokens, se utiliza en la siguiente etapa del compilador, llamada el analizador sintáctico. Los tokens son como bloques de construcción que ayudan a construir la estructura general del programa. Entender cómo se conectan las cosas en el código es clave para que el programa funcione bien.

## Reconocimiento basado en Gramáticas Regulares:

Para encontrar esos tokens, usamos algo llamado expresiones regulares. Son como patrones que describen diferentes tipos de cosas en el código, como nombres, operadores o palabras clave. Esto ayuda a que el análisis léxico sea consistente con las reglas del lenguaje de programación.

## Generación de Componentes Léxicos:

Lo que obtienes al final del análisis léxico son componentes léxicos o símbolos. Estos representan concretamente los tokens que encontramos. Son súper importantes para las etapas posteriores del compilador, como el análisis sintáctico y la creación de un código intermedio. Si estos componentes no son precisos, podríamos tener problemas más adelante.

### Primera Fase del Compilador:

Así que, básicamente, el análisis léxico es la primera cosa que hace el compilador. Toma el código fuente, lo divide en trozos manejables y crea una especie de "resumen" que se pasa al analizador sintáctico. Hacerlo bien desde el principio es crucial porque cualquier error aquí podría causar problemas en las etapas posteriores.

### Conclusión:

En resumen, el análisis léxico es un paso clave en la construcción de programas. Ayuda a organizar la complejidad del código, asegurando que sea entendido correctamente y se ejecute sin problemas. Los analizadores léxicos necesitan ser precisos para que todo el proceso de compilación funcione bien.