



ESTRUCTURA DE UN COMPILADOR

FASES DE UN COMPILADOR

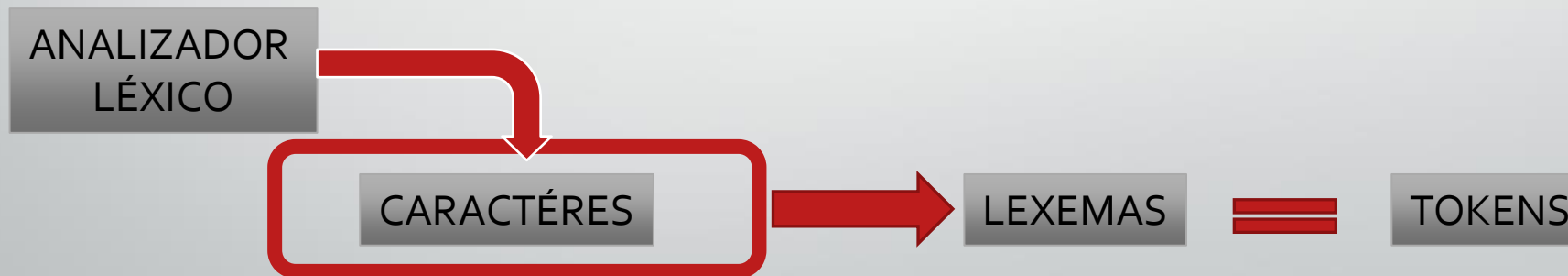
- El compilador, al ser un programa que traduce código legible por el ser humano a código legible por la computadora, necesita pasar el contenido del código por una serie de procesos de verificación, conversión y optimización. Estos procesos o fases son:
 - Análisis Léxico
 - Análisis Sintáctico
 - Análisis Semántico
 - Generación de código intermedio
 - Optimización
 - Generación de código Objeto



ANÁLISIS LÉXICO

ANALIZADOR LEXICO

- PRIMERA FASE DE UN COMPILADOR.
- SU PRINCIPAL TAREA ES LEER LOS CARACTERES DEL PROGRAMA FUENTE, LOS AGRUPA EN LEXEMAS Y GENERA UNA SECUENCIA DE TOKENS.



**PROGRAMA
FUENTE**

**ANALIZADOR
LÉXICO**

TOKEN

**ANALIZADOR
SINTÁCTICO**

**ANALISIS
SEMANTICO**

OBTENER SIGUIENTE TOKEN

**TABLA DE
SIMBOLOS**

FUNCIONES ANALIZADOR LEXICO

Como el analizador léxico es la parte del compilador que lee el texto de origen, debe realizar otras tareas aparte de identificar lexemas.

Una de esas tareas es eliminar los comentarios y el *espacio en blanco* (*espacios, nuevas líneas, tabuladores, separadores de tokens*).

Otra de las tareas es correlacionar los mensajes de error generados por el compilador con el programa fuente.

TOKENS, PATRONES Y LEXEMAS

Al hablar de un análisis léxico, utilizamos tres términos distintos, pero relacionados:

- **Token**: es un símbolo abstracto que representa un tipo de unidad léxica.
- **Patrón**: descripción de la forma que pueden tomar los lexemas de un token.
- **Lexema**: secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token.

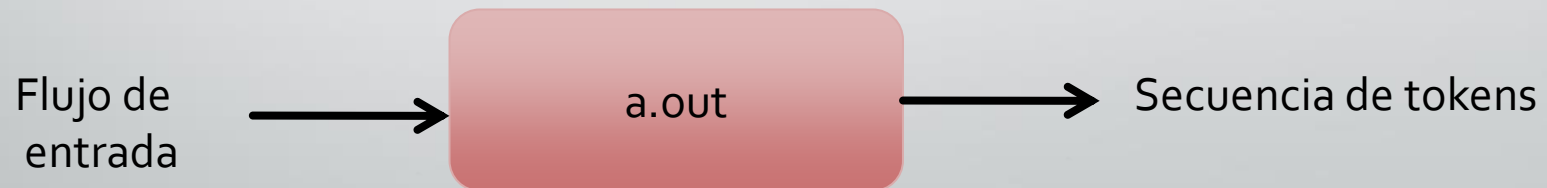
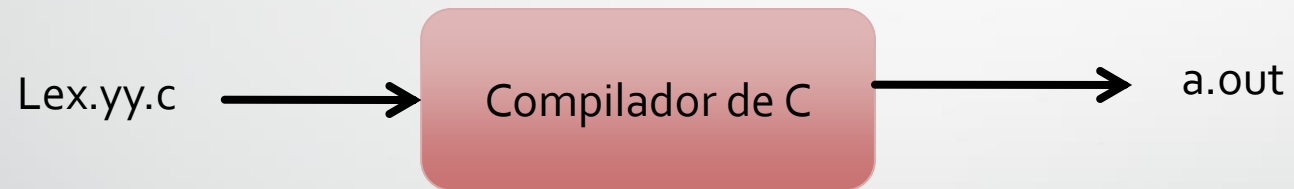
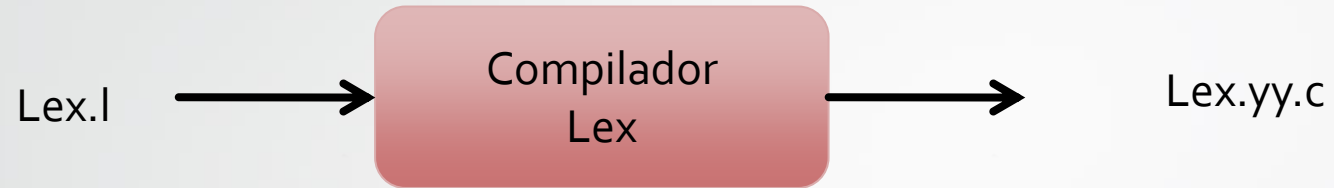


EL GENERADOR DE ANALIZADORES LÉXICOS LEX

EL GENERADOR DE ANALIZADORES LÉXICOS LEX

- El generador de código léxico Lex (o Flex) es una herramienta que nos permite especificar un analizador léxico mediante la especificación de expresiones regulares para describir patrones de los tokens

EL GENERADOR DE ANALIZADORES LÉXICOS LEX



Estructura de los programas en Lex

- Un programa en Lex tiene la siguiente forma:

Declaraciones

%%

Reglas de traducción

%%

Funciones auxiliares

La sección de declaraciones tiene las declaraciones de variables, **constantes de manifiesto** (**identificadores que se declaran para representar una constante**) y definiciones regulares.

Cada una de las reglas de traducción tiene la siguiente forma:

Patrón {Acción}

Cada patrón es una expresión regular, la cual puede usar las definiciones regulares de la sección de declaraciones. Las acciones son fragmentos de código, generalmente escritos en C.

La tercera sección contiene las funciones adicionales que se utilizan en las acciones.

