



Laboratorio de Programación de Sistemas

PRÁCTICA 02

ANALIZADOR LÉXICO-SINTÁCTICO ENSAMBLADOR SIC ESTÁNDAR

Analizador léxico y sintáctico del ensamblador de SIC STD

Objetivo

El alumno implementará una **aplicación** que realice el **análisis léxico y sintáctico** para el lenguaje **ensamblador** de la arquitectura **SIC** en su versión **estándar**.

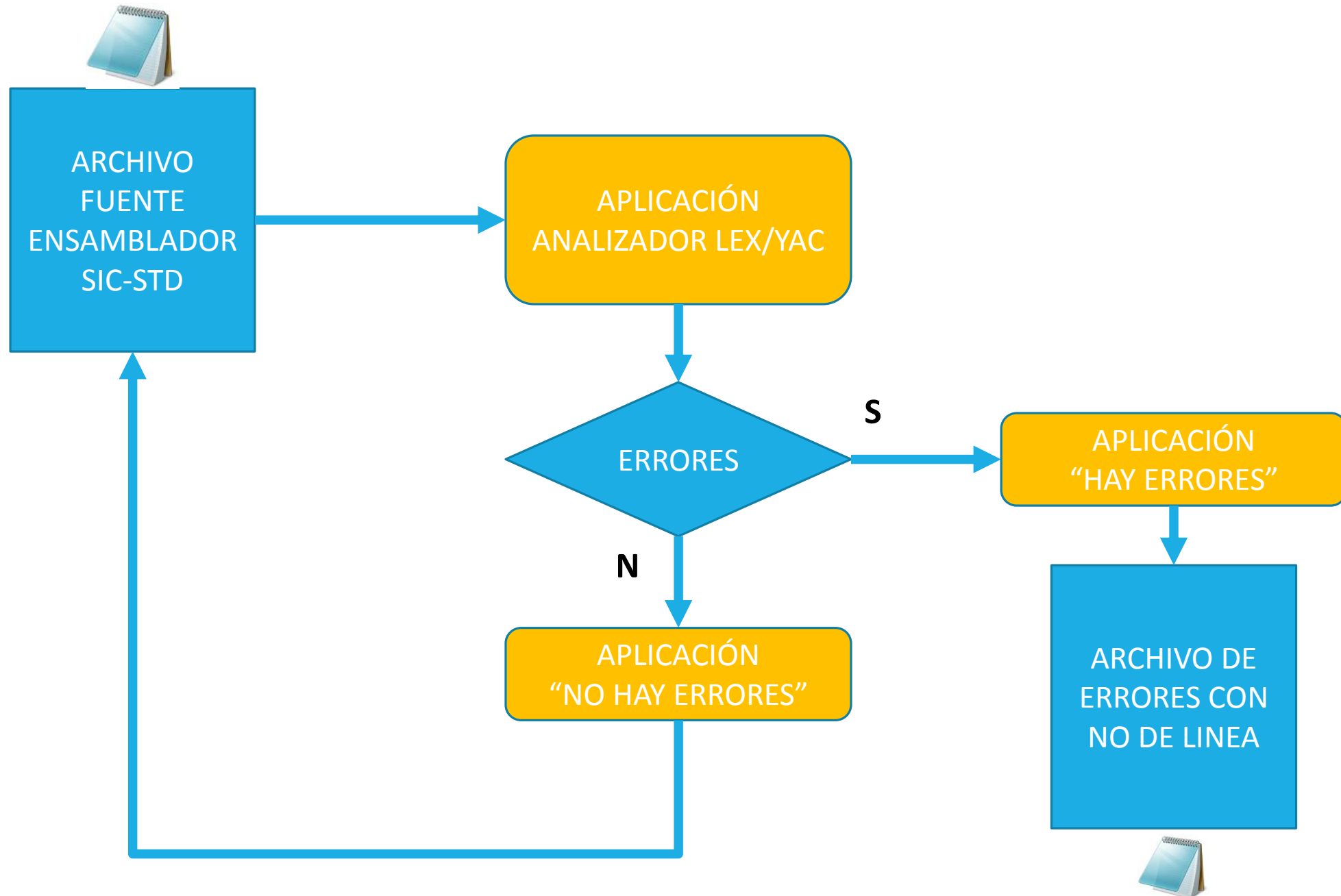
Antecedentes teóricos

- a) Análisis léxico
- b) Análisis sintáctico
- c) Lenguaje ensamblador SIC estándar

El análisis léxico y sintáctico fue cubierto ampliamente en la materia de Compiladores e Intérpretes A y la estructura de los programas en lenguaje ensamblador para la SIC estándar se estudia en las primeras semanas de la materia de Programación de Sistemas.

Desarrollo

- 1) Definir los componentes léxicos y requeridos por lenguaje ensamblador SIC estándar.
- 2) Definir las reglas gramaticales necesarias para representar la estructura sintáctica del lenguaje ensamblador SIC estándar.
- 3) Crear una aplicación para implementar los componentes léxicos y las reglas gramaticales definidas previamente. La aplicación deberá tomar como **entrada un programa fuente del ensamblador SIC estándar y producir como salida un archivo de texto con los errores del programa.**
- 4) El archivo fuente de entrada deberá utilizar la extensión .s y el archivo de texto de salida deberá utilizar el mismo nombre del archivo de entrada pero con la extensión .t
- 5) El archivo de texto de salida deberá ser colocado en el mismo directorio donde se ejecute la aplicación.
- 6) Los ejemplos de prueba no deberán ser modificados para adaptarse a sus reglas gramaticales, sino que al contrario la gramática deberá diseñarse de acuerdo a ellos.
- 7) Los ejemplos de prueba se podrán modificar para ocasionar errores intencionales.
- 8) Los errores léxicos/sintácticos deberán informar del número de línea (aproximado) donde se encuentra el error en el archivo fuente.



Reporte

1. Descripción de los componentes léxicos definidos.
2. Descripción de las reglas gramaticales utilizadas.
3. Descripción del manejo de archivos tanto de entrada como de salida.
4. Conclusiones y posibles mejoras.

Definición dirigida por la sintaxis

programa	→	inicio proposiciones fin
inicio	→	etiqueta START NUM FINL proposición
Fin	→	END entrada FINL END entrada
entrada	→	ID e
proposiciones	→	proposiciones proposición proposición
proposición	→	Instrucción Directiva
instrucción	→	etiqueta CODOP opinstruccion FINL
directiva	→	etiqueta tipodirectiva opdirectiva FINL
tipodirectiva	→	TIPOBYTE TIPOWORD RESB RESW
etiqueta	→	ID e
opinstruccion	→	ID indexado e
indexado	→	INDICE e
opdirectiva	→	NUM CONSTHEX CONSTCAD

