

Laboratorio de Programación de Sistemas

PRÁCTICA IV – PASO 2 DEL ENSAMBLADOR PARA SIC ESTÁNDAR

Paso 2 Ensamblador SIC - STD

❑ **Objetivo**

Ensamblar instrucciones y generar registros objeto de la SIC/STD.

❑ **Antecedentes teóricos**

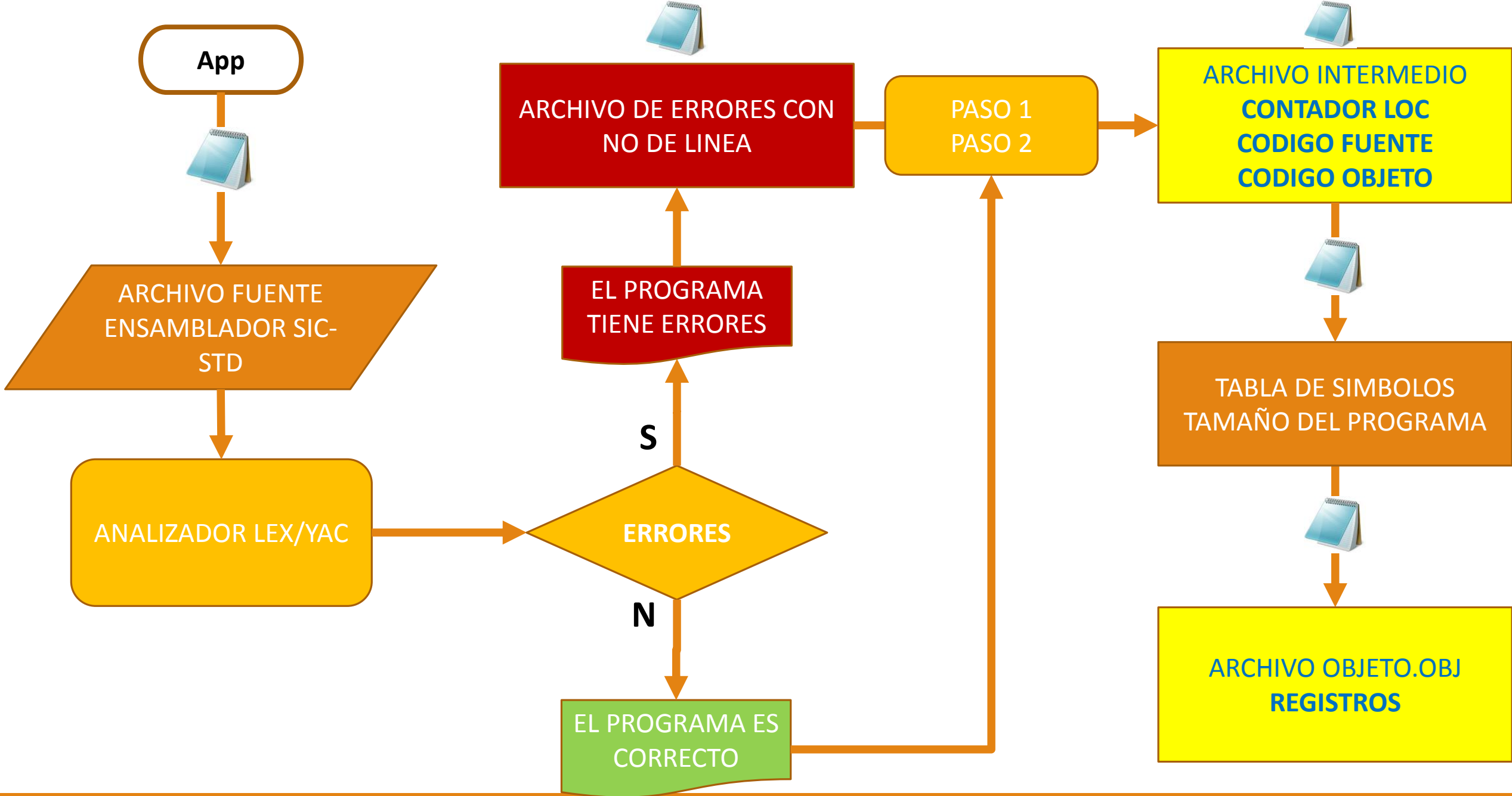
- a) Conjunto de instrucciones
- b) Formato de instrucciones
- c) Modos de direccionamiento
- d) Operaciones AND, OR, SHR, SHL a nivel de bits (bitwise operators)
- e) Registros objeto

❑ **Requisitos previos**

- ❖ Paso 1 del ensamblador
- ❖ Tabla de símbolos

Desarrollo

1. Definir una tabla para almacenar el valor numérico de los códigos de operación.
2. Generar el *registro de encabezado*.
3. *Ensamblar* las *instrucciones* utilizando operaciones a nivel de bits.
4. *Ensamblar* las directivas **BYTE** y **WORD** traduciendo su valor a *hexadecimal*.
5. Colocar el código ensamblado dentro de los *registros de texto*.
6. Generar los *registros de texto* al encontrar alguna **directiva RESB, RESW** o cuando estén *llenos*.
7. Generar el *registro de fin*.
8. En caso de que no exista una etiqueta dentro del programa, *indicar el error* y *ensamblar* la instrucción utilizando una dirección **-1 (0xFFFF)**
9. Agregar en el *archivo intermedio* la columna de *código objeto*.





ARCHIVO INTERMEDIO CONT.LOC - CODIGO FUENTE - CODIGO OBJETO

	CP	Etiqueta	Instrucción	Operando	Codigo Objeto
1	001000	COPY	START	1000	---
2	001000	FIRST	STL	RETADR	141033
3	001003	CLOOP	JSUB	RDREC	482039
4	001006		LDA	LENGTH	001036
5	001009		COMP	ZERO	281030
6	00100C		JEQ	ENDFIL	301015
7	00100F		JSUB	WRREC	482060
8	001012		J	CLOOP	3C1003
9	001015	ENDFIL	LDA	EOF	00102A
10	001018		STA	BUFFER	0C1039
11	00101B		BYTE	X'123	Error: Sintaxis
12	00101B		LDA	THREE	00102D
13	00101E		STA	LENGTH	0C1036
14	001021		JSUB	WRREC	482060
15	001024		LDL	RETADR	081033
16	001027		RSUB		4C0000
17	00102A	BYTE	BYTE	C'11'	Error: Sintaxis
18	00102A	EOF	BYTE	C'EOF'	454F46
19	00102D	THREE	WORD	000003	000003
20	001030	ZERO	WORD	000000	000000



TABLA DE SIMBOLOS TAMAÑO DEL PROGRAMA

Simbolo	Direccion
FIRST	1000
CLOOP	1003
ENDFIL	1015
EOF	102A
THREE	102D
ZERO	1030
RETADR	1033
LENGTH	1036
HOLA	1039
OTRO	1039

Tamaño del programa: 001079H



ARCHIVO OBJETO.OBJ REGISTROS

```

HCOPY 001000001079
T0010001E1410334820390010362810303010154820603C100300102A0C1
T00101E150C10364820600810334C0000454F46000003000000
T0020391E041030001030E0205D30203FD8205D28103030A0575490392CA
T0020571C1010364C0000F1001000041030E02078302063329039DC20782
T002072073820634C000005
E001000

```

Reporte

- 1) Describir la estructura utilizada para la tabla de códigos de operación.
- 2) Describir el procedimiento utilizado para:
 - a) ensamblar instrucciones
 - b) ensamblar directivas BYTE y WORD
 - c) generar los registros de encabezado, texto y fin
- 3) Describir los problemas que se presentaron durante el desarrollo de la práctica y explicar cómo fueron resueltos.
- 4) Redactar sus conclusiones y proponer posibles mejoras.

```

1 void Ensamblador::paso2()
2 {
3     leer_primera_linea_entrada(archivo_intermedio);
4     if (CODOP == "START")
5     {
6         escribir_linea_listado();
7         leer_siguiete_linea_entrada();
8     }
9     escribir_registro_encabezado_programa_objeto();
10    asigar_valor_inicial_primer_registro_texto();
11    while(CODOP != "END")
12    {
13        if (linea_actual_no_es_comentario())
14        {
15            buscar CODOP en TABOP;
16            if (se encuentra)
17            {
18                if (linea_actual_tiene OPERANDO)
19                {
20                    busca OPERANDO en TABSIM;
21                    if (se encuentra)
22                    {
23                        guardar_el_valor_del_simbolo_como_direccion_del_operando();
24                    }
25                    else
26                    {
27                        almacena 0 como direccion del operando;
28                        activar_bandera_error (simbolo indefinido);
29                    }
30                }
31                else
32                {
33                    almacena 0 como direccion del operando;
34                    ensamblar_instruccion_codigo_objeto();
35                }
36            }
37            else if (CODOP == "BYTE" || COPOP == "WORD")
38            {
39                convertir_constante_en_codigo_objeto();
40                if (codigo_objeto_no_cabe_en_registro_texto_actual())
41                {
42                    escribir_registro_texto_programa_actual();
43                    asignar_valor_inicial_nuevo_registro_texto();
44                }
45                añadir_codigo_objeto_al_registro_de_texto();
46            }
47        }
48        //fin no es comentario
49    }
50    //fin paso2
51    //fin no es comentario
52    escribir_una_linea_del_listado();
53    leer_siguiete_linea_entrada();
54    //fin while not END
55    escribir_ultimo_registro_texto_programa_objeto();
56    escribir_registro_fin_programa_objeto();
57    escribir_ultima_linea_listado();
58    //fin paso2

```