



UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

ÁREA DE CIENCIAS DE LA COMPUTACIÓN

“LABORATORIO PROGRAMACIÓN DE SISTEMAS”

**PRACTICA 10:
“Paso 1 – SIC Extendida”**

**PROFESOR:
ING. AGUSTÍN HERNÁNDEZ GARCÍA**

ALUMNO: Braulio Alejandro García Rivera

CLAVE ÚNICA: 239196

FECHA: 26/11/2020

SEMESTRE: 2020-2021/I

1. INTRODUCCIÓN

En este documento se explica cómo fue la generación de una gramática con la herramienta ANTLR4 para abrir dos archivos con extensión .sx y evaluar si tenía errores gramaticales.

2. INSTALACIÓN Y CONFIGURACIÓN

En Visual Studio 2017 se instalaron las siguientes extensiones:

- Antlr4Code
- ANTLR Language Support

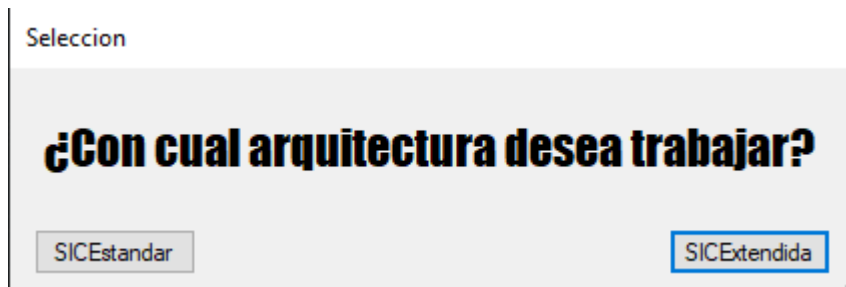
También se instalaron las bibliotecas de Antlr4 como paquetes NuGet:

- Antlr4 v4.6.6
- Antlr4.CodeGenerator v4.6.6
- Antlr4.Runtime v4.6.6

La configuración se realizó en base a la explicación del profesor y los ejemplos de la plataforma didacTIC.

3. INTERFAZ

Se agrego un formulario para seleccionar Sic estándar y Sic Extendida.



Selección

¿Con cual arquitectura desea trabajar?

SICEstandar SICExtendida

Form1

ARQUITECTURA: ☐ SIC Estándar ☒ SIC Extendida

NUEVO MODIFICAR

ABRIR ANALIZAR PASO 2

GUARDAR TAMAÑO DEL PROGRAMA: MAPA DE MEMORIA

TAMAÑO DEL ARCHIVO: 0

LECTURA

```

1 PRUEBASX START 4000H
2 FIRST +LDX #ZERO
3 LDA ZERO
4 LOOP ADD @TABLE,X
5 TX COUNT
6 +JLT LOOP
7 STA TOTAL
8   RSUB
9 TABLE RESW 2000
10 COUNT RESW 1
11 ZERO WORD 0
12 TOTAL RESW 1
13 END FIRST

```

ARCHIVO INTERMEDIO

#	CP	ETIQUETA	INSTRUCCION	DIRECCION	ERROR
1	4000	PRUEBASX	START	4000H	NO
2	4000	FIRST	+LDX	#ZERO	NO
3	4000		LDA	ZERO	NO
4	4003	LOOP	ADD	@TABLE,X	NO
5	4006		TX	COUNT	NO
6	4009		+JLT	LOOP	NO
7	4009		STA	TOTAL	NO
8	400C		RSUB		NO
9	400F	TABLE	RESW	2000	NO
10	577F	COUNT	RESW	1	NO
11	5782	ZERO	WORD	0	NO

TABSIM

SÍMBOLO	DIRECCION
FIRST	4000
LOOP	4003
TABLE	400F
COUNT	577F
ZERO	5782
TOTAL	5785

ERRORES

ARCHIVO OBJ

4. IMPLEMENTACIÓN GRAMÁTICA SIC EXTENDIDA

Para la generación de la gramática para la SIC Extendida se requirió apoyarse sobre la gramática de la SIC Estándar modificándola para la funcionalidad correcta como dividir la instrucción en los formatos posibles que son (formatouno | formatodos | formatotres | formatocuatro), también se agregaron nuevas terminales y los registros que utiliza la SIC Extendida: ('A'|'X'|'L'|'B'|'S'|'T'|'F'), también se agregaron los direccionamientos ('@'|'#').

5. SOPORTE A LA COMPATIBILIDAD

Para que se soportara el análisis de archivos para la SIC Estándar y la SIC Extendida se dividió la clase en cargada de ensamblar en dos cada una a su respectiva arquitectura, en la clase principal se utilizó una bandera para verificar que tipo de arquitectura se seleccionó para trabajar.

6. SOPORTE PARA LA TABLA DE CÓDIGOS DE OPERACIÓN

Para que se soportara la tabla de códigos de operación se creó la segunda gramática donde se reconocieran los símbolos que nos permitieran conocer el modo de direccionamiento y concretar de manera satisfactoria el paso 1.

7. IMPLEMENTACIÓN PASO 1 SIC EXTENDIDA

Analiza línea e código, mientras exista alguna.

Divide el código fuente en líneas.

Analizador sintáctico para la línea.

Muestra el archivo intermedio.

Muestra TABSIM.

Muestra errores.

8. PROBLEMAS

Un problema fue el momento de generar la gramática ya que había que analizar detalladamente para dividir los formatos que se utilizaron y ver la manera de estructurarla de manera correcta.

9. POSIBLES MEJORAS

Mejorar el análisis al momento de encontrar errores para saber con certeza donde se está produciendo y estructurar una descripción más concreta o específica sobre la gramática.

10. CONCLUSIÓN

En esta práctica lo más importante fue la correcta implementación de la gramática para empezar a procesar las instrucciones de manera que permitiera evaluar los archivos para iniciar con la estructuración del CP y poder detectar errores para evaluarlos de forma correcta.