

ANÁLISIS LÉXICO

>> Análisis del lenguaje <<					
Tipo	Concepto general	Representación	Contenido	Trabajo que desempeña	Objetivo del trabajo
Alfabeto	Elementos que en conunto conforman todos los caracteres aceptados por el lenguaje en cuestión	Σ	Σ = {Def, def, Barras, Pie, titulo, ejex, ejey, etiquetas, valores, unir, tipo, Cantidad, total, Porcentaje, extra, Ejecutar, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ;, :, {, }, ,, [,], +, -, *, /, (,), #}	Listar los elementos que son aceptados por el lenguaje	Distinguir entre entradas aceptadas por el lenguaje que luego serán clasificardas como tokens y distinguir las entradas inválidas que procederán a ser ignoradas
Palabra reservada	Única combinación posible resultante de un patrón impuesto en el lenguaje	DEF, BARRAS, PIE, TITULO, EJEX, EJEY, ETIQUETAS, VALORES, UNIR, TIPO, CANTIDAD, TOTAL, PORCENTAJE, EXTRA, EJECUTAR	DEF = Def, def; BARRAS = Barras; PIE = Pie; TITULO = titulo; EJEX = ejex; EJEY = ejey; ETIQUETAS = etiquetas; VALORES = valores; UNIR= unir; TIPO = tipo; CANTIDAD = Cantidad; TOTAL = total; PORCENTAJE = Porcentaje; EXTRA = extra; EJECUTAR = Ejecutar;	"Reservar" el uso de la combinación de símbolos que le corresponde	Representar un lexema expecífico esperado en el parser
Definiciones regulares	Agrupación con nombre que permite que al monento de requerir los elementos del conjunto, se utilice el nombre que se le ha asignado, con el fin de mejorar la legibilidad de las especificacioens léxicas	digito	digito = [0-9]	Hacer referencia a los elemtos del conjunto dígito en la expresión regular que requiera de ellos, en este caso, la expresión regular numero, que permite que se acepten números enteros y decimales	Mejorar la legibilidad de la especificación léxica
Expresión regular	Regla que emplea operadores de expresiones regulares y símbolos o elementos del alafabeto del lengujae, con el fin de permitir combinaciones realizadas a partir de cierto patrón designado por ésta	simbolosAceptados	".","","[""{""}"]","[""[""]"]"	Permitir que el lenguaje reconozca como una cadena válidad aquella que esté conformada por únicamente uno de los elementos que se encuentran en esta expresión	Clasificar los lexemas con el ID correspondiente al lexema para luego ser enviados al parser y así éste último pueda utilizarlo como un Terminal que pueda ser empleado por la(s) regla(s) de producción que lo necesite(n).
		operadores	"+" " "- "*" " /" "(") ")"	Permitir que el lenguaje reconozca como una cadena válidad aquella que esté conformada por únicamente uno de los elementos que se encuentran en esta expresión	Clasificar los lexemas con el ID correspondiente al lexema para luego ser enviados al parser y así éste último pueda utilizarlo como un Terminal que pueda ser empleado por la(s) regla(s) de producción que lo necesite(n).
		numero	{digito}+(".{digito}+)?	Permitir que el lenguaje reconozca como una cadena válidad aquella que esé conformada por dígitos y forme una cantidad entera o con decimales	Clasificar los lexemas con el ID = número para luego ser enviados al parser y así éste último pueda utilizarlo como un Terminal que pueda ser empleado por la(s) regla(s) de producción que lo necesite(n).
		finDeLinea	\r \n \r\n	Permitir que el lenguaje reconozca la cadena que representa a cualquiera de las formas en las que pueda ser representado un final de línea	Reconocer los lexemas para utilizarlos como auxiliar para formar otros formar otros tokens o patrones de expresiones regulares, o ignorarlos para que el lexer no haga envíos innecesarios al parser
		tabulacion	[\t\f]	Permitir que el lenguaje reconozca la cadena que representa a cualquiera de las formas en las que pueda ser representado una tabulación	Reconocer los lexemas para utilizarlos como auxiliar para formar otros formar otros tokens o patrones de expresiones regulares, o ignorarlos para que el lexer no haga envíos innecesarios al parser
		espacioEnBlanco	{finDeLinea} {tabulacion}	Permitir que el lenguaje reconozca la cadena que representa a cualquiera de las formas en las que pueda ser representado un espacio en blanco	Reconocer los lexemas para utilizarlos como auxiliar para formar otros formar otros tokens o ignorarlos para que el lexer no haga envíos innecesarios al parser
		cuerpoComentario	[^\r\n]	Permitir que el lenguaje reconozca cualquier cantidad de caracteres diferentes al salto de línea o salto de línea y retorno de carro	Permitir que el cuerpo de un comentario de línea sea cualquier cosa excepto aquel o aquellos caracteres que permiten ir a una línea más abajo
		comentario	"#" {cuerpoComentario}* {finDeLinea}?	Permitir que el lenguaje acepte comentarios de línea, que inicien con un "#", en cualquier parte de la entrada	Permtir que al ser reconocida esta estructura por el lexer, no sea enviada al parser, puesto que esta combinación no es necesaria para ninguna de las reglas de producción que definen la gramática perteneciente al lenguaje

		secuencias de escape [literales]	<code>\\t \\n \\r \\\\" \\</code>	Permitir que el lenguaje reconozca el conjunto de caracteres escritos por el usuario que en conjunto forman la expresión literal de las secuencias de escape más comunes	Permitir que si al estar en el estado léxico CADENA el usuario ingresa la representación literal de las secuencias de escape más conocidas, el analizador léxico pueda concatenarlas literalmente al lexema del token cadena
		"cualquier caracter"	<code>[^]</code>	Permitir que el lenguaje reconozca cualquier caracter que no sea la cadena vacía	Enviar al estado ERROR cuando un caracter no definido en alguna de las definiciones y expresiones regulares o no pertenezcann al inicio de cualquiera de las palabras resrvadas del lenguaje, se pueda concatenar cada uno de los caracteres que aparezcan antes de un espacio en blanco y después del que dio entrada a este estado léxico
Estado léxico [estructura propia de JFLEX]	Estructura que permite crear ámbitos diferentes y con ello reconocer conjuntos de expresiones regulares diferentes a las que se analizan en otro estado	CADENA	<code><CADENA> { [^\\n\\r\\\"\\\\]+ \\t \\n \\r \\\\" \\ }</code>	Permitir que se formen tokens del tipo cadena, que forman cadenas aceptadas por el lenguaje	Permitir que al entrar a la definición del estado CADENA, el analizador léxico, solo verifique aquellas expresiones regualres que serán útiles para formar cadenas del tipo "cadena"
		ERROR	<code><ERROR>{ {espacioEnBlanco} }</code>	Permitir que se formen tokens del tipo error, que forma engloba las cadenas no aceptadas por el lenguaje	Permitir que al entrar a la definición del estado ERROR, todos los caracteres recibidos se concatenten hasta que se encuentre un caracter en blanco, lo cual indica en este lenguaje que el conjunto de caracteres posibles a reunir, se ha "cerrado"

ANÁLISIS SINTÁCTICO

>> Gramática formal <<				
G = {NT, T, RP, So}				
No Terminales [NT]	Terminales [T]	Reglas de Producción [RP]		Estado Inicial [So]
		No.	Regla	
INICIO	NUMERO	1	INICIO -> 1erSECCION 2daSECCION	INICIO
1erSECC	CADENA			
DEFINICION	SUMA	2	1erSECCION -> 1erSECCION DEFINICION 1erSECCION -> DEFINICION	
GRAFICO	RESTA			
CUERPO_BARRA	MULTI	3	DEFINICION -> DEF GRAFICO	
PARAM_BARRA	DIV			
ATRIB_GENERAL	:	4	GRAFICO -> BARRA { CUERPO_BARRA } GRAFICO -> PIE { CUERPO_PIE }	
TUPLA	{			
ATRIB_BARRA	}	5	CUEPRO_BARRA -> PARAM_B1 ... PARAM_B4	
CUERPO_PIE	:			
PARAM_PIE	,	6	PARAM_BARRA -> ATRIB_BARRA ;	
ATRIB_PIE	(
EL_TIPO)	7	TUPLA -> TUPLA , { VAL_NUMER , VAL_NUMER } TUPLA -> { VAL_NUMER , VAL_NUMER }	
LISTA_CADENAS	[
LISTA_NUMEROS]	8	ATRIB_BARRA -> ATRIB_GENERAL ATRIB_BARRA -> EJEX ; [LISTA_CADENAS] ATRIB_BARRA -> EJEY ; [LISTA_NUMEROS]	
VAL_NUMER	DEF			
2daSECC	BARRAS	9	CUERPO_PIE -> PARAM_PIE1 ... PARAM_PIE6 CUERPO_PIE -> PARAM_PIE1 ... PARAM_PIE7	
EXE	PIE			
	TITULO	10	PARAM_PIE -> ATRIB_PIE ;	
	EJEX			
	EJEY	11	ATRIB_PIE -> ATRIB_GENERAL ATRIB_PIE -> ETIQUETAS ; [LISTA_CADENAS] ATRIB_PIE -> VALORES ; [LISTA_NUMEOS] ATRIB_PIE -> EXTRA ; CADENA ATRIB_PIE -> EL_TIPO ATRIB_PIE -> TOTAL ; VAL_NUMER	
	ETIQUETAS			
	VALORES			
	UNIR			
	TIPO			
	TOTAL			
	CANTIDAD	12	EL_TIPO -> CANTIDAD EL_TIPO -> PORCENTAJE	
	PORCENTAJE			
	EXTRA	13	LISTA_CADENAS -> LISTA_C , CADENA LISTA_CADENAS -> CADENA	
	EJECUTAR			
		14	LISTA_NUMEROS -> LISTA_N , VAL_NUMER LISTA_NUMEROS -> VAL_NUMER	
		15	VAL_NUMER -> VAL_NUMER SUMA VAL_NUMER VAL_NUMER -> VAL_NUMER RESTA VAL_NUMER VAL_NUMER -> VAL_NUMER MULTI VAL_NUMER VAL_NUMER -> VAL_NUMER DIV VAL_NUMER VAL_NUMER -> (VAL_NUMER) VAL_NUMER -> NUMERO	
		16	2daSECC -> 2daSECC EXE 2daSECC -> EXE	
		17	EXE -> EJECUTAR (CADENA) ;	

ANÁLISIS SINTÁCTICO

>> Análisis del lenguaje <<				
No.	Regla de Producción	Estructura(s) que analiza	Objetivo	Error reconocido
1	INICIO -> 1erSECCION 2daSECCION INICIO -> 2daSeccion	El esqueleto del archivo de entrada	Obligar al usuario a que el archivo de entrada que creará posea la primer sección [de definición], antes de que la segunda sección aparezca	Sección (1 o 2) mal estructurada
2	1erSECCION -> 1erSECCION DEFINICION 1erSECCION -> DEFINICION	La primer sección de la definición de un archivo de entrada para la aplicación de graficación	Obligar que en el archivo de entrada al menos aparezca una vez la estructura para definir los gráficos a ejecutar	-
3	DEFINICION -> DEF GRAFICO	La estructura general de la sección de definición de gráficos	Obligar a que la estructura de definición de gráficos en el archivo de entrada inicie con la palabra reservada del token DEF y continúe con la demás estructura para crear un gráfico que varía de contenido según el tipo	-
4	GRAFICO -> BARRA { CUERPO_BARRA } GRAFICO -> PIE { CUERPO_PIE }	La estructura específica de la definición, según el tipo de grafico a generar	Obligar a que luego del nombre del tipo de gráfica, se encierre entre llaves el cuerpo de la definición correspondiente a la gráfica	Estructura de definición de gráfico incorrecta
5	CUEPRO_BARRA -> PARAM_B1 ... PARAM_B4	El listado de parámetros de la gráfica de barras	Especificar al analizador que el listado únicamente debe constar de 4 parámetros para recibir los atributos que definirán la gráfica a realizar	Número de parámetros del gráfico de barras incorrecto
6	PARAM_BARRA -> ATRIB_BARRA ;	Estructura general de los atributos a listar en el cuerpo de la definición del gráfico	Especifcar que el único formato en el que pueden ser declarados los atributos, en este caso del gráfico de barras, es colocando la estructura del atributo seguida de ";"	Punto y coma esperado pero no recibido
7	TUPLA -> TUPLA , { VAL_NUMER , VAL_NUMER } TUPLA -> { VAL_NUMER , VAL_NUMER }	El tipo de contenido que el atributo unir posee	Especificar que el contenido de unir consta de al menos una pareja de valores numéricos separada por una coma y encerrada entre llaves	Pareja de tupla mal formada
8	ATRIB_BARRA -> ATRIB_GENERAL ATRIB_BARRA -> EJEX : [LISTA_CADENAS] ATRIB_BARRA -> EJEY : [LISTA_NUMEROS]	El listado de atributos que específicamente la gráfica barra debe poseer	Listar los atributos de los que una grádica de tipo barra está conformada para ser creada correctamente	Estructura de atributo de gráfico de barras incorrecto
9	CUERPO_PIE -> PARAM_PIE1 ... PARAM_PIE6 CUERPO_PIE -> PARAM_PIE1 ... PARAM_PIE7	El listado de parámetros de la gráfica de pie	Especificar al analizador que la gráfica de pie debe ser definida por 6 o 7 atributos para ser creada corerctamente	Número de parámetros del gráfico de pie incorrecto
10	PARAM_PIE -> ATRIB_PIE ;	Estructura general de los atributos a listar en el cuerpo de la definición del gráfico	Especifcar que el único formato en el que pueden ser declarados los atributos, en este caso del gráfico de pie, es colocando la estructura del atributo seguida de ";"	Punto y coma esperado pero no recibido
11	ATRIB_PIE -> ATRIB_GENERAL ATRIB_PIE -> ETIQUETAS : [LISTA_CADENAS] ATRIB_PIE -> VALORES : [LISTA_NUMEOS] ATRIB_PIE -> EXTRA : CADENA ATRIB_PIE -> TIPO EL_TIPO ATRIB_PIE -> TOTAL : VAL_NUMER	El listado de atributos que específicamente la gráfica barra debe poseer	Listar los atributos de los que una gráfica de tipo pie está conformada para ser creada correctamente	Estructura de atributo de gráfico de pie incorrecto
12	EL_TIPO -> CANTIDAD EL_TIPO -> PORCENTAJE	El contenido del atributo "tipo"	Listar los tipos de cantidades que puede representar el total de un gráfico de pie	Tipo de toal inválido
13	LISTA_CADENAS -> LISTA_C , CADENA LISTA_CADENAS -> CADENA	El listado de cadenas	Especificar que un listado de cadenas debe constar con al menos una de ellas y en caso de haber más, que estén separadas por ","	Estructura de lista de cadenas incorrecta
14	LISTA_NUMEROS -> LISTA_N , VAL_NUMER LISTA_NUMEROS -> VAL_NUMER	El listado de valores numéricos	Indicar que en un listado de valores numéricos debe existir al menos un dato y en caso de haber más que se separaren con ","	Estructura de lista de números incorrecta
15	VAL_NUMER -> VAL_NUMER SUMA VAL_NUMER VAL_NUMER -> VAL_NUMER RESTA VAL_NUMER VAL_NUMER -> VAL_NUMER MULTI VAL_NUMER VAL_NUMER -> VAL_NUMER DIV VAL_NUMER	Operaciones aritméticas básicas	Indicar los tipos de operaciones aritméticas que son aceptadas por el lenguaje y que puede estar conformada por al menos una expresión que puede ser agrupada por medio de paréntesis	-

	VAL_NUMER -> { VAL_NUMER } VAL_NUMER -> <u>NUMERO</u>			
16	2daSECC -> 2daSECC EXE 2daSECC -> EXE	La segunda sección de la definición de un archivo de entrada para la aplicación de graficación	Obligar que en el archivo de entrada al menos aparezca una vez la estructura para ejecutar un gráfico previamente definido	-
17	EXE -> <u>EJECUTAR</u> { <u>CADENA</u> } ;	La estructura de la instrucción ejecutar	Especificar que una instrucción ejecutar, debe constar de la palrba reservada ejecutar y un cuerpo que contenga la cadena que indique el gráfico a generar seguido de un ";" para indicar el fin de la instrucción	-