

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 = {&, {, {, }, }, }, /, *, , , =, >, <, !, dígito, letra, +, -, , , identificador, import, new, this, public, protected, private, final, class, int, boolean, String, char, double, Object, void, true, false, if, else, for, while, switch, case, default, break, return }	Listar los elementos que son aceptados por el lenguaje	Distinguir entre entradas aceptadas por el lenguaje que luego serán clasificadas 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	IMPORT, NEW, THIS, PUBLIC, PROTECTED, PRIVATE, FINAL, CLASS, INT, BOOLEAN, STRING, CHAR, DOUBLE, OBJECT, VOID, TRUE, FALSE, IF, ELSE, FOR, WHILE, SWITCH, CASE, DEFAULT, BREAK, RETURN	{import, new, this, public, protected, private, final, class, int, boolean, String, char, double, Object, void, true, false, if, else, for, while, switch, case, default, break, return }	"Reservar" el uso de la combinación de símbolos que le corresponde	Representar un lexema específico esperado en el parser	
Definiciones regulares	Agrupación con nombre que permite que al momento de requerir los elementos del conjunto, se utilice el nombre que se le ha asignado, con el fin de mejorar la legibilidad de las especificaciones léxicas	dígito	dígito: [0-9] letra: [a-zA-Z]	Hacer referencia a los elementos del conjunto dígito en la expresión regular que requiera de ellos, en este caso, la expresión regular número, que permite que se acepten números enteros y decimales, y para el caso de la otra, hacer referencia a las letras de las que consta el alfabeto	Mejorar la legibilidad de la especificación léxica	
Expresión regular	Regla que emplea operadores de expresiones regulares y símbolos o elementos del alfabeto del lenguaje, con el fin de permitir combinaciones realizadas a partir de cierto patrón designado por ésta	numero	numero = (dígito)+(".(dígito)+)?	Permitir que el lenguaje reconozca como una cadena válida aquella que esté conformada por dígitos y forme una cantidad entera o con decimales	Permitir que el parser reciba nombrados los conjuntos de palabras que cumplen el patrón de NUMERO, para que pueda reconocer e identificar si cumple con lo esperado	
		caracter	caracter = ""[letra]]""	Permitir que el lenguaje reconozca como una cadena válida aquella que esté conformada por letras o espacios vacíos	Permitir que el parser reciba nombrados los conjuntos de palabras que cumplen el patrón de CHARACTER, para que pueda reconocer e identificar si cumple con lo esperado	
		identificador	identificador = ("\$(letra))("_(letra))("\$(letra)){"(dígito)}+	Permitir que el lenguaje reconozca como una cadena válida aquella que esté conformada por letras, dígitos, guiones bajos y símbolos de dólar, en el patrón especificador, para ser contados como identificadores	Permitir que el parser reciba nombrados los conjuntos de palabras que cumplen el patrón de IDENTIFICADOR, para que pueda reconocer e identificar si cumple con lo esperado	
		finDeLinea	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	Permitir que el lexer al reconocer los los conjuntos de palabras que cumplen el patrón de FIN DE LÍNEA, proceda a ignorarlos debido a que no se requieren para el lenguaje actual	
		tabulacion	tabulacion = [\f]	Permitir que el lenguaje reconozca la cadena que representa a cualquiera de las formas en las que pueda ser representado una tabulación	Permitir que el lexer al reconocer los los conjuntos de palabras que cumplen el patrón de TABULACIÓN, proceda a ignorarlos debido a que no se requieren para el lenguaje actual	
		espacioEnBlanco	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	Permitir que el lexer al reconocer los los conjuntos de palabras que cumplen el patrón de ESPACIO EN BLANCO, proceda a ignorarlos debido a que no se requieren para el lenguaje actual	
		ComentarioMultiLinea	ComentarioMultiLinea = {comentarioTradicional} {comentarioDocumentacion}	Permitir que el lenguaje reconozca los diferentes tipos de comentarios multilinea	Permitir que el lexer identifique los conjuntos de palabras que cumplen el patrón de COMENTARIO MULTILÍNEA, para que pueda reconocer e identificar si cumple con lo esperado	
		corpoComentario	corpoComentario = [^\r\n]	Permitir que el lenguaje acepte comentarios de línea, que inicien con un "#", en cualquier parte de la entrada	Permitir que el lexer identifique los conjuntos de palabras que cumplen el patrón de CUERPO DE COMENTARIO, para que pueda ser recopilada la información necesaria para hacer la comparación	
		comentarioLinea	comentarioLinea = "/" {corpoComentario}* {finDeLinea}?	Permitir que el lenguaje acepte los típicos comentarios de línea, que inician con un "/", en cualquier parte de la entrada	Permitir que el lexer identifique los conjuntos de palabras que cumplen el patrón de un COMENTARIO DE LÍNEA, para que pueda ser recopilada la información necesaria para hacer la comparación	
		comentarioTradicional	comentarioTradicional = "/" [^] -"/" "/"***"/	Permitir que el lenguaje acepte comentarios multilinea tradicionales	Permitir que el lexer identifique los conjuntos de palabras que cumplen el patrón de un COMENTARIO TRADICIONAL, para que pueda ser recopilada la información necesaria para hacer la comparación	

			comentarioDocumentacion	comentarioDocumentacion = "/"** {contenidoComentario}***+ "/"	Permitir que el lenguaje acepte comentarios multilinea semejantes a los de Java _doc, utilizados para documentar	Permitir que el lexer identifique los conjuntos de palabras que cumplen el patrón de un COMENTARIO DE DOCUMENTACIÓN, para que pueda ser recopilada la información necesaria para hacer la comparación
			contenidoComentario	contenidoComentario = ({[^*] \'+ [^/'] }*)	Permitir que el lenguaje tome como contenido todo aquel que cumpla con la expresión dada	Permitir que el lexer identifique los conjuntos de palabras que cumplen el patrón de un CONTENIDO DE COMENTARIO, para que pueda ser recopilada la información necesaria para hacer la comparación
	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	STRING	<CADENA> { [^\n\r\"\\]+ \\t \\n \\r \\\\" \\\\ }	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 regulares que serán útiles para formar cadenas del tipo "cadena"
			ERROR	<ERROR> { {espacioEnBlanco} }	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"