"2019, Año del Caudillo del Sur, Emiliano Zapata"

DEPARTAMENTO DE SISTEMAS COMPUTACIONALES E INFORMÁTICA

ASUNTO: SOLICITUD DE ACTIVIDADES

Celaya, Guanajuato, 17/Septiembre/2019

LENGUAJES Y AUTÓMATAS II
DOCENTE DESIGNADO: ISC. RICARDO GONZÁLEZ GONZÁLEZ
SEMESTRE AGOSTO-DICIEMBRE 2019

ACTIVIDAD 2 (VALOR 70 PUNTOS)

LEA CUIDADOSAMENTE, Y REALICE LAS SIGUIENTE ACTIVIDADES, CONSIDERANDO LOS CRITERIOS DE CALIDAD PROPUESTOS EN LOS DOCUMENTOS DE LA GUÍA TUTORIAL, Y LA RÚBRICA DE EVALUACIÓN,

EL LECTOR DEBE TOMAR MUY EN CUENTA QUE ESTA ACTIVIDAD ES UN EXAMEN, Y NO DE UNA TAREA SENCILLA, PUES DEMANDA TIEMPO PARA INVESTIGAR, LEER, ANALIZAR, REDACTAR, ILUSTRAR Y PROPONER DE MANERA PROFESIONAL LOS TEMAS PROPUESTOS EN LA ESTRUCTURA TEMÁTICA DE ESTA MATERIA.

1. TOMANDO EN CUENTA TODAS LAS INDICACIONES Y EXPLICACIONES EN CLASE Y UNA VEZ DEFINIDA LA GRAMÁTICA Y EL LENGUAJE PROTOTIPO A IMPLEMENTAR, COMO EQUIPO INVESTIGUEN, DISEÑEN E IMPLEMENTEN LA PRIMERA FASE O ETAPA DE UN ANALIZADOR LÉXICO.

2. LA ACTIVIDAD COMO EQUIPO DEBERÁ COMENZAR CON LA INVESTIGACIÓN Y FUNDAMENTACIÓN DE LOS SIGUIENTES TEMAS.

   A. INVESTIGAR. ¿ QUÉ ES UN ANÁLISIS LÉXICO APLICADO A LA VALORACIÓN DE UN LENGUAJE FORMAL ?
   B. INVESTIGAR. ¿ EN QUÉ CONSISTE UN ANÁLISIS LÉXICO Y QUÉ LO CARACTERIZA?
   C. IDENTIFICAR. ¿ QUÉ CASOS DE ESTUDIOS SON LOS IMPORTANTES A CONSIDERAR EN EL ANÁLISIS LÉXICO ?
   D. INVESTIGAR Y PROPONER. ¿ QUÉ PROCESOS Y PROBLEMAS ATIENDE UN ANÁLISIS LÉXICO ?
   E. INVESTIGAR. ¿ CÓMO IMPLEMENTAR UTILIZANDO UN ANÁLISIS LÉXICO ?

   NOTA : ESTOS TEMAS DEBEN SER EL RESULTADO DE UNA INVESTIGACIÓN Y ANÁLISIS COMO EQUIPO, Y NO UNA TRANSCRIPCIÓN DE TEMAS AISLADOS, PUES EN TODO MOMENTO LAS FUENTES DE CONSULTA DEBEN SER LA BASE DEL DESARROLLO DE ESTE PUNTO Y SUS APARTADOS.

3. COMO EQUIPO Y DERIVADO DEL ANÁLISIS ANTERIOR SE DEBEN PROPONER LOS ALGORITMOS Y ESTRUCTURAS DE DATOS NECESARIAS PARA LA IMPLEMENTACIÓN DE UN PROTOTIPO DE ANALIZADOR LÉXICO. ES DECIR, MANEJO Y OPERACIONES CON ARCHIVOS, TABLA DE SÍMBOLOS, DISEÑO DE AUTÓMATAS, PILA DE ERRORES, ETC.

   CONCRETAMENTE EN ESTE PUNTO DEBERÁN COMO EQUIPO, REDACTAR Y DETALLAR TODOS LOS ELEMENTOS QUE SE USARÁN PARA LA IMPLEMENTACIÓN DEL ANALIZADOR LÉXICO.

4. PARA EL INCISO C DEL PUNTO 2 ANTERIOR, SE DEBERÁN CREAR PROGRAMAS DE MÍNIMO 25 LÍNEAS (SIN CONSIDERAR LOS COMENTARIOS) CADA UNO, EN LOS CUALES SE CODIFIQUE EN EL LENGUAJE PROTOTIPO, INSTRUCCIONES CON LÓGICA QUE EJEMPLIFIQUEN LOS ERRORES QUE EN CADA CASO DE ESTUDIO SE PROPONGAN.

   TALES PROGRAMAS DEBEN ESTAR PERFECTAMENTE DOCUMENTADOS Y CO-RELACIONADOS CON LOS CASOS DE ESTUDIO CORRESPONDIENTES.

5. CARACTERÍSTICAS QUE LA ACTIVIDAD 2 DEBE POSEER PARA CONSIDERARSE COMPLETA.

   A. EL FUNDAMENTO DE LA GRAMÁTICA A UTILIZAR, ASÍ COMO SU DEFINICIÓN FORMAL Y EL ALFABETO A UTILIZAR.

   B. LA CARACTERIZACIÓN DEL LENGUAJE PROTOTIPO, ES DECIR SU DESCRIPCIÓN MEDIANTE NOTACIÓN BNF. SE DEBEN INCLUIR ADEMÁS LOS SÍMBOLOS ESPECIALES COMO DELIMITADORES IMPLÍCITOS Y EXPLÍCITOS, OPERADORES LÓGICOS, RELACIONALES Y ARITMÉTICOS.

   C. CATEGORIZACIÓN E IDENTIFICACIÓN EN POR ID, DE CADA UNO LOS ELEMENTOS QUE EL LENGUAJE PROTOTIPO PROPONE.

   D. PLANTEAMIENTO Y FUNDAMENTACIÓN DE LOS CASOS DE USO DEL ANALIZADOR LÉXICO, ASÍ COMO LOS ERRORES EN QUE DERIVARÁ CADA UNO DE ELLOS.

   E. PREPARACIÓN DE LOS PROGRAMAS SUFICIENTES, ESCRITOS EN EL LENGUAJE PROTOTIPO, QUE APOYEN LA COMPROBACIÓN DE CADA CASO DE ESTUDIO. TALES PROGRAMAS DEBERÁN ESTAR COMPLETAMENTE DOCUMENTADOS.

   F. GENERACIÓN DE UN CATÁLOGO DE ERRORES, CORRELACIONADO A LOS CASOS DE USO PROPUESTOS.

   G. DISEÑO DE UNA SOLUCIÓN MODELADA EN OBJETOS, APOYADA EN DIAGRAMAS UML QUE DESCRIBAN LA ARQUITECTURA PROPUESTA PARA EL PROTOTIPO DEL ANALIZADOR LÉXICO.

   H. PROPUESTA Y MODELADO DE LAS PROPIEDADES Y COMPORTAMIENTOS DE UNA TABLA DE SÍMBOLOS, ASÍ COMO LA JUSTIFICACIÓN DE CADA UNA DE LAS COLUMNAS QUE LA INTEGREN.

"2019, Año del Caudillo del Sur, Emiliano Zapata"

I. PROPUESTA Y MODELADO DE LAS PROPIEDADES Y COMPORTAMIENTOS DE UNA PILA DE ERRORES.

J. MODELADO DE LOS PROCESOS DE APERTURA Y LECTURA DEL ARCHIVO DE CÓDIGO FUENTE, ASÍ COMO DEL PROCESO DE TOKENIZACIÓN (QUE NO SE DEBERÁ IMPLEMENTAR A USANDO FUNCIONES DE BIBLIOTECAS DE TERCEROS O EXTERNAS).

K. DISEÑO, PRUEBAS, MODELADO E IMPLEMENTACIÓN DE LOS AUTÓMATAS SUFICIENTES Y NECESARIOS PARA LA CATEGORIZACIÓN DE CADA UNO DE LOS TOKENS GENERADOS.

L. MODELADO E IMPLEMENTACIÓN DE UN PROCESO DE :

LECTURA DE CÓDIGO FUENTE => TOKENIZACIÓN => CATEGORIZACIÓN DE LOS TOKENS => CONSTRUCCIÓN Y LLENADO DE LA TABLA DE SÍMBOLOS => MANEJO Y DESPLIEGUE DE ERRORES => DESPLIEGUE DE LA TABLA DE SÍMBOLOS RESULTANTE.

M. GENERACIÓN DE UN CALENDARIO DE ACTIVIDADES PLANIFICADAS VS. ACTIVIDADES REALIZADAS.

N. GENERACIÓN DE UNA BITÁCORA DE INCIDENCIAS.

O. TODAS LAS EVIDENCIAS GENERADAS Y REUNIDAS DEBERÁN INTEGRARSE A UN ARCHIVO PDF, NOMBRADO COMO SE INDICA MÁS ADELANTE.

ESTAS EVIDENCIAS PODRÁN SE ELABORAR CON HERRAMIENTAS ELECTRÓNICAS, COMO PROCESADOR DE TEXTO, DE IMÁGENES, HOJAS DE CÁLCULO, ETC.

P. EL NÚCLEO DE CADA ALGORITMO CODIFICADO TAMBIÉN DEBERÁ FORMAR PARTE DEL ARCHIVO DE EVIDENCIAS.

Cuan Abril Ant

"2019, Año del Caudillo del Sur, Emiliano Zapata"

NOTA A CONSIDERAR.

CADA UNO DE LOS PUNTOS ANTERIORES DEBE SER DESARROLLADO CON LA PROFUNDIDAD ACORDE A UN NIVEL PROFESIONAL, Y APEGÁNDOSE COMPLETAMENTE A LAS DIRECTRICES DE LA GUÍA TUTORIAL.

NO CONCIBA ESTE TRABAJO, COMO UN SIMPLE RESUMEN O EJERCICIO DE TRANSCRIPCIÓN, PUES EL VALOR INDICADO AL INICIO DE ESTA ACTIVIDAD LE DARÁ A USTED UNA BUENA IDEA DE LO QUE SE ESPERA DE ELLA, EN CUANTO A CALIDAD Y EL APRENDIZAJE OBTENIDO, MISMO QUE SERÁ PUESTO A PRUEBA MEDIANTE UN EXAMEN ESCRITO O BIEN ORAL EN CLASE.

SI DECIDIÓ ELABORAR ESTA ACTIVIDAD EN EQUIPO, CADA INTEGRANTE DE ÉSTE DEBERÁ POSEER EL MISMO NIVEL DE CONOCIMIENTO, PUES TAN SOLO REPARTIR TEMAS ENTRE LOS INTEGRANTES DEL EQUIPO, SUPONDRÍA UN GRAVE ERROR DE INTERPRETACIÓN A LA INTENCIÓN DIDÁCTICA REAL DE ESTA ACTIVIDAD.

POR ÚLTIMO, ESTA ACTIVIDAD SOLO SE PODRÁ DESARROLLAR EN EQUIPO, SI SE REGISTRÓ EN UNO PREVIAMENTE, UTILIZANDO EL FORMATO ENTREGADO EN LA ACTIVIDAD INICIAL. DE LO CONTRARIO DEBERÁ ELABORAR Y ENTREGAR LA ACTIVIDAD DE FORMA INDIVIDUAL.

LA ENTREGA DE DICHO REGISTRO SE HARÁ VÍA CORREO ELECTRÓNICO ENVIANDO ÉSTE AL PROFESOR DESIGNADO, Y POSTERIORMENTE EN CLASE ENTREGANDO LA HOJA EN FÍSICO.

NOTA GENERAL DE LA ACTIVIDAD :

SE DEBE CONSIDERAR Y TOMAR EN CUENTA PARA EL CORRECTO CUMPLIMIENTO DE ESTA ACTIVIDAD, LO SOLICITADO EN LA GUÍA TUTORIAL, CONCRETAMENTE EN EL PUNTO 3 INCISO i (*Trabajo en equipo*).

"2019, Año del Caudillo del Sur, Emiliano Zapata"

OBSERVACIONES:

- ✓ LA REVISIÓN SERÁ EN DIVERSAS VERTIENTES. PUEDE SER AL MOMENTO DE SOLICITAR LA CARPETA DE EVIDENCIAS, O BIEN PUEDE SER AL SOLICITAR LA EXPOSICIÓN DE LA TAREA EN LA CLASE. TAMBIÉN PUEDE SER POR SOLICITUD EXPRESA DEL INTERESADO A PARTICIPAR EN CLASE EXPONIENDO BREVEMENTE SU ACTIVIDAD.
- ✓ AQUELLAS ACTIVIDADES EN FORMATO DIGITAL SE DEBERÁN TENER SIEMPRE, Y EN TODO MOMENTO A LA MANO EN UNA MEMORIA USB.
- ✓ ÉSTAS ACTIVIDADES PODRÁN SER SOLICITADAS EN LA CLASE, O BIEN PARA SU ENVÍO A UNA CUENTA DE CORREO.
- ✓ ESTAS ACTIVIDADES DEBEN ESTAR LISTAS E INTEGRADAS A LA CARPETA DE EVIDENCIAS (*FÍSICAMENTE*) A LA FECHA DE ENTREGA INDICADA AL FINAL DE ÉSTE DOCUMENTO.
- ✓ CADA HOJA QUE ENTREGUE DE SU ACTIVIDAD, DEBERÁ ESTAR FIRMADA AL MARGEN DERECHO, INCLUIDA LA PROPIA SOLICITUD DE LA ACTIVIDAD.
- ✓ UNA VEZ ELABORADA SU ACTIVIDAD, RECUERDE DIGITALIZARLA Y NOMBRARLA EN BASE A LA NOMENCLATURA QUE SE INDICA MÁS ADELANTE EN ESTE DOCUMENTO.
- ✓ SI SUS EVIDENCIAS ENVIADAS POR CORREO, NO CUMPLEN CON LA NOMENCLATURA SOLICITADA, NO SERÁN CONSIDERADAS COMO EVIDENCIAS PARA SU EVALUACIÓN.
- ✓ CON ESTA ACTIVIDAD, USTED DEBERÁ IR INTEGRANDO SUS CARPETAS FÍSICA Y ELECTRÓNICA DE EVIDENCIAS, Y AL FINAL DEL SEMESTRE EN UN DISCO COMPACTO HARÁ ENTREGA DE SU CARPETA ELECTRÓNICA DE EVIDENCIAS.
- ✓ PARA TENER DERECHO A LA REVISIÓN Y EVALUACIÓN DE SUS ACTIVIDADES, DEBE REGISTRAR SU ASISTENCIA A CLASE, EL DÍA SEÑALADO PARA LA ENTREGA DE LA MISMA.
- ✓ FALTAR A CLASE EL DÍA DE LA ENTREGA, ANULA LA REVISIÓN DE SUS EVIDENCIAS.
- ✓ POR ÚLTIMO, POR FAVOR GESTIONE APROPIADAMENTE SU TIEMPO, Y SEA PUNTUAL EN SU ENTREGA Y ASÍ EVITAR PROBLEMAS DE NULIDAD POR EXTEMPORANEIDAD.
- ✓ AÚN PARA TRABAJOS EN EQUIPO APLICAN TODAS LAS MISMAS OBSERVACIONES ANTERIORES.
- ✓ MUY IMPORTANTE: SI EN LA REVISIÓN DE LA ACTIVIDAD SE ENCUENTRAN PLAGIOS EN FORMA DE TRANSCRIPCIONES TOTALES O PARCIALES DE INFORMACIÓN EN DOCUMENTOS YA EXISTENTES, LA ACTIVIDAD SE ANULARÁ TOTALMENTE.
- ✓ PARA QUE ESTA ACTIVIDAD PUEDA ALCANZAR EL VALOR DE PUNTAJE MÁS ALTO, DEBERÁ DARSE UNA COPIOSA E IMPORTANTE PARTICIPACIÓN EN CLASE DURANTE TODO EL PERIODO DE EVALUACIÓN, A FIN DE CONVALIDAR EL APRENDIZAJE ADQUIRIDO.

"2019, Año del Caudillo del Sur, Emiliano Zapata"

LA NOMENCLATURA SOLICITADA PARA ENVIAR SU TRABAJO ES LA SIGUIENTE :

```
AAAA-MM-DD_MATERIA_DOCUMENTO_EQUIPO_NOCTROL_APELLIDOS_NOMBRE_SEM.PDF
```

(NOTA : *** TODO EN MAYÚSCULA ***)

DONDE :

AAAA	:	AÑO
MM	:	MES
DD	:	DÍA
MATERIA	:	LAII, PB, PE, TSO
DOCUMENTO	:	A1-ACTIVIDAD 1, P1-PRACTICA 1, R1-REPORTE 1, T1-TAREA 1, PG1-PROGRAMA, ETC. (CAMBIANDO EL NÚMERO CONSECUTIVO POR EL QUE CORRESPONDA)
EQUIPO	:	NÚMERO DEL EQUIPO QUE CORRESPONDA SEGÚN INDICACIÓN DEL PROFESOR.
NOCTROL	:	SU NÚMERO DE CONTROL
APELLIDOS	:	SUS APELLIDOS
NOMBRE	:	SU NOMBRE
SEM	:	EL PERIODO SEMESTRAL EN CURSO: ENE-JUN / AGO-DIC

EJEMPLO :

SI EL TRABAJO SE HACE EN EQUIPO.

```
2019-09-27_LAII_A2_EQUIPO_99_9999999_PEREZ_PEREZ_JUAN_AGO-DIC19.PDF
```

SI EL TRABAJO SE HACE INDIVIDUALMENTE.

```
2019-09-27_LAII_A2_9999999_PEREZ_PEREZ_JUAN_AGO-DIC19.PDF
```

"2019, Año del Caudillo del Sur, Emiliano Zapata"

FECHA DE ENTREGA:

VÍA CORREO ELECTRÓNICO, EL VIERNES 27 DE SEPTIEMBRE DEL 2019, CON HORA LÍMITE DE ENTREGA HASTA LAS 14:00 HORAS (2 DE LA TARDE).

EN CASO DE QUE EL TRABAJO SE HAYA ELABORADO EN EQUIPO, EL JEFE DEL MISMO SERÁ EL ÚNICO RESPONSABLE DE ENVIAR LA ACTIVIDAD A LA SIGUIENTE CUENTA DE CORREO:

ricardo.gonzalez@itcelaya.edu.mx

MUY IMPORTANTE:

DESPUÉS DE LAS 14:00 HRS. EN PUNTO, LA ACTIVIDAD YA SERÁ CONSIDERADA COMO EXTEMPORÁNEA Y NO CONTARÁ COMO EVIDENCIA PARA SU EVALUACIÓN.

SE LE SUGIERE ENVIAR CON ANTICIPACIÓN SU ACTIVIDAD A FIN DE EVITAR CONFLICTOS POR NO ENTREGAR ÉSTA A TIEMPO.

RECUERDE ANEXAR A SU ARCHIVO .PDF DE EVIDENCIAS, ESTA SOLICITUD DE ACTIVIDADES CON TODAS LAS HOJAS FIRMADAS EN EL MARGEN DERECHO.

POR ÚLTIMO, CONSIDERE EL SIGUIENTE CALENDARIO OFICIAL, PARA GESTIONAR ADECUADAMENTE EL TIEMPO QUE SE LE OTORGA PARA DESARROLLAR ESTA ACTIVIDAD.

CALENDARIO 2019-2020 - EVALUACIÓN 2

¿QUÉ ES UN ANÁLISIS LÉXICO APLICADO A LA VALORACIÓN DE UN LENGUAJE FORMAL?

De una manera simple, el **análisis léxico**, es un programa que convierte el código fuente en **lexemas**. Esto lo hace: mediante la lectura del código carácter por carácter, reconoce los lexemas y tiene como salida una secuencia de **tokens** que representan los lexemas.

Un **lexema**, es una cadena que corresponde a un patrón, algunos ejemplos, son las palabras reservadas, identificadores, operadores, entre otros.

Un **token** es una forma de representar un lexema, puede tener distintos componentes, como el tipo (operador, identificador, etc.), valor del lexema, la línea en que fue encontrado; esto, según las necesidades del compilador.

Ejemplo:

Token	Lexema	Descripción
if	if	Palabra reservada
oprel	<, >, <=, >=, ==, !=	Operadores de comparación
id	i, nombre, var	Secuencia de letras y dígitos
num	5, 10, 2000	Constante numérica

¿EN QUÉ CONSISTE UN ANÁLISIS LÉXICO Y QUÉ LO CARACTERIZA?

Un analizador léxico lee carácter a carácter el programa fuente y los agrupa para formar tokens que pueden representar:

- Palabras reservadas
- Identificadores
- Operadores
- Símbolos especiales
- Constantes numéricas
- Constates de caracteres

El analizador sintáctico realiza la petición para que se realice el análisis léxico conforme va avanzando en la gramática. Cuando se realiza esta petición, se hace la lectura hasta que se encuentre el próximo componente léxico.

Además de las funciones anteriores, algunas funciones consideradas como secundarias, pero igual de importante en la realización de un analizador léxico son:

- Manejo del archivo de código fuente (apertura, lectura, cierre y manejo de errores).
- Eliminar comentarios, espacios en blanco, tabuladores y saltos de línea.
- Incluir ficheros
- Expandir macros y funciones
- Contar las líneas

¿QUÉ PROCESOS Y PROBLEMAS ATIENDE UN ANÁLISIS LÉXICO?

Los procesos y problemas principales son:

- Identificar tokens
- Agregar tokens a la tabla de símbolos
- Eliminar espacios en blanco y comentarios
- Relacionar mensajes de error con el código
- Expandir macros, si existen
- Leer caracteres del código fuente

¿CÓMO IMPLEMENTAR UTILIZANDO UN ANÁLISIS LÉXICO?

- **Identificar tokens.** Cuando se encuentra un lexema, se ingresa en varios autómatas y el autómata que lo acepte nos devuelve el Token ID que hayamos designado.
- **Agregar tokens a la tabla de símbolos.** El análisis léxico lo único que hace, es agregar los símbolos que va encontrando, sin ninguna información en especial. La información sobre el tipo de variable, las líneas donde se hace referencia, su valor, etc. son agregadas por el analizador sintáctico.
- **Eliminar espacios en blanco y comentarios**. Esto lo hace leyendo los caracteres hasta que ninguno de estos se encuentre, es decir, si se encuentra un espacio en blanco se lee el siguiente carácter y así hasta encontrar otro símbolo que no sea un símbolo en blanco.
- **Relacionar mensajes de error con el código.** Los errores léxicos, se reducen principalmente a uno: error en un token. Es decir, que no sea válido en ninguna de las categorías que definimos.
- **Expandir macros (si existen).** En el caso de que existan macros, el trabajo del analizador léxico, es "copiar y pegar" la macro en donde ha sido llamado.
- **Leer caracteres del código fuente.** Se lee carácter a carácter hasta encontrar algún delimitador implícito (espacio, tabulador o salto de línea) o un cambio de contexto (paréntesis, punto y coma, operador relacional, operador de asignación, etc.).

ALGORITMOS Y ESTRUCTURAS DE DATOS NECESARIAS PARA LA IMPLEMENTACIÓN DE UN PROTOTIPO DE ANALIZADOR LÉXICO (MANEJO Y OPERACIONES CON ARCHIVOS, TABLA DE SÍMBOLOS, DISEÑO DE AUTÓMATAS, PILA DE ERRORES, ETC.).

Manejo de archivos

El algoritmo básico para leer de archivos secuenciales es:

1. Abrir el archivo
2. Leer caracter por caracter hasta llegar al final del archivo
3. Cerrar el archivo

Tabla de símbolos

Se hace uso de una estructura de datos conocida como Hash Table a la que se le ingresa una llave y un valor, a la llave le aplica una función Hash y con esto, obtiene la dirección de memoria donde se encuentra el valor.

Para este caso, como llave, se usará el lexema encontrado y como valor una clase personalizada llamada Token, que contendrá el tipo (palabra reservada o id), una etiqueta numérica y la línea donde fue declarado.

Autómatas

Para los autómatas, se realizarán clases que simulen el funcionamiento de un Autómata Finito Determinista, por lo que se requerirá de:

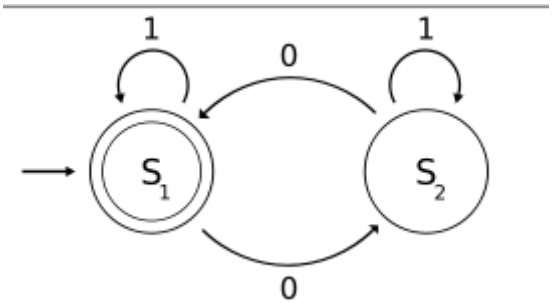
- Reglas de producción
- Estado inicial
- Estados finales
- Alfabeto

Para esto, haré uso de las siguientes clases:

- **KeyPair**. Hará la función de un TDA. Será la forma de buscar en la lista de reglas, el siguiente estado.
- **Rules**. Contendrá toda la información de las reglas en una estructura conocida como HashMap, que tiene un comportamiento similar a un HashTable. Mediante un objeto KeyPair, sabremos cual es el siguiente estado del autómata.
- **Control**. Simula la cabeza lectora del autómata. Sabrá en que estado se encuentra actualmente el autómata, y cual es el siguiente.
- **Automaton**. Simula un autómata finito determinista. Nos dirá si una cadena se encuentra o no en el lenguaje definido.

Ejemplo

Para ejemplificar mejor el funcionamiento de estas clases, a continuación se muestra una pequeña explicación, se basa en el autómata simple:



Se conoce que

$$M =< (Q, \Sigma, \delta, q_0, F >$$
$$\text{Donde:}$$
$$Q = (1, 2)$$
$$\Sigma = (0, 1)$$
$$\delta = ($$
$$\delta(1, 0) = 2,$$
$$\delta(1, 1) = 1,$$
$$\delta(2, 0) = 1,$$
$$\delta(2, 1) = 2,$$
$$)$$
$$q_0 = 1$$
$$F = 1$$

Pila de errores

Como su nombre lo indica, requiere de una estructura de pila, también conocida como LIFO (*Last In First Out*). Su función principal es insertar valores numéricos que se encuentran en un catálogo de errores, para que se muestren dichos errores.

LENGUAJE PROTOTIPO EN NOTACIÓN BNF. SE DEBEN INCLUIR ADEMÁS LOS SÍMBOLOS ESPECIALES COMO DELIMITADORES IMPLÍCITOS Y EXPLÍCITOS, OPERADORES LÓGICOS, RELACIONALES Y ARITMÉTICOS.

```
<programa> ::= { declare <decls> begin <inst>* }
<decls> ::= <decl><decls> | ε
<decl> ::= <tipo> <id>;
<tipo> ::= int | float | boolean | char
<inst> ::=    print(<id>);
            | read(<id>);
            | if (<bool>) { <inst>* }
            | if (<bool>) { <inst>* } else { <inst>* }
            | while (<bool>) { <inst>* }
            | do { <inst>* } while (<bool>);
            | break;
            | <asig>
<asig> ::= <id> = <expr>;
<bool> ::= <bool> or <comb> | <comb>
<comb> ::= <comb> and <comb> | <comp>
<comp> ::= <expr> <oprel> <expr> | <expr>
<oprel> ::= < | > | <= | >= | == | !=
<expr> ::=   <expr> + <term>
           | <expr> - <term>
           | <term>
<term> ::=   <term> * <val>
           | <term> / <val>
           | <val>
<val> ::= <num> | <real> | true | false | <cadena>
<num> ::= [0-9]+
<real> ::= [0-9]+.[0-9]+
<cadena> ::= "[a-zA-Z0-9,.-_]*"
<id> ::= [a-z]([a-zA-Z0-9])*
```

CATEGORIZACIÓN E IDENTIFICACIÓN DE CADA UNO LOS ELEMENTOS DEL LENGUAJE PROTOTIPO.

Palabras reservadas

Valor	ID
declare	200
begin	201
print	203
read	204
if	205
else	206
while	207
do	208
or	209
and	210
true	217
false	218
break	219
int	220
float	221
boolean	222
char	223

Identificadores

Nombre	ID
id	224

Numeros

Nombre	Regex	ID
int_val	[0-9]+	226
float_val	[0-9]+.[0-9]+	227

Operadores relacionales

Nombre	Valor	ID
lt	<	211
gt	>	212
le	<=	213
ge	>=	214
eq	==	215
ne	!=	216

Cadenas

Nombre	Regex	ID
char_val	"[a-zA-Z0-9.,_-]*"	225

Elementos unarios

Nombre	Valor	ID
co	{	226
cc	}	227
po	(	228
pc	)	229
op_arit	[+-*/]	230
asig	=	231
semi_colon	;	232

CATÁLOGO DE ERRORES.

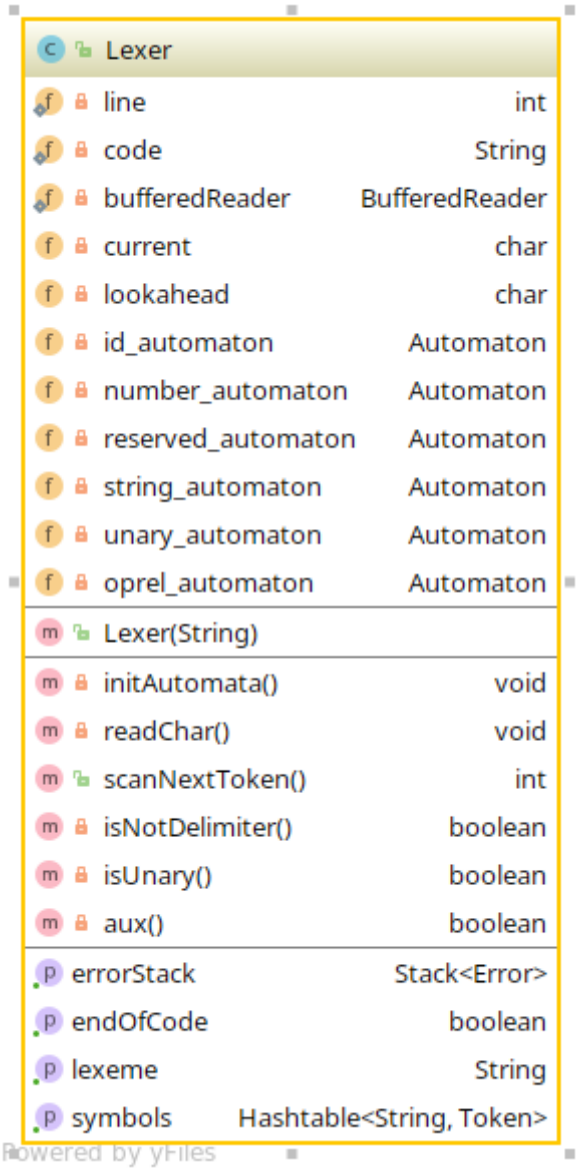
- **100**. Error de token desconocido (no es reconocido como un id, numero, palabra reservada, operador relacional o elemento unario).

SOLUCIÓN MODELADA EN OBJETOS, APOYADA EN DIAGRAMAS UML QUE DESCRIBAN LA ARQUITECTURA PROPUESTA PARA EL PROTOTIPO DEL ANALIZADOR LÉXICO.

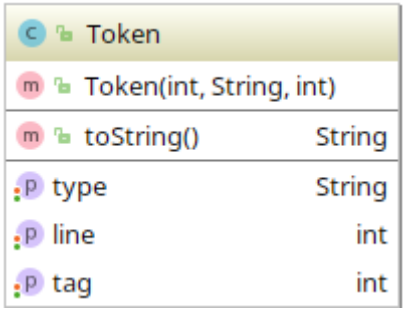
Package lexer

Clase Lexer

NOTA IMPORTANTE. A comparación del documento enviado el viernes 27 de septiembre, a la clase *Lexer*, se agregó el método *isUnary()*, para hacer más fácil una sentencia *if*. Además se agregó el campo *lexeme* para poder obtener desde otras clases el lexema que se acaba de leer.

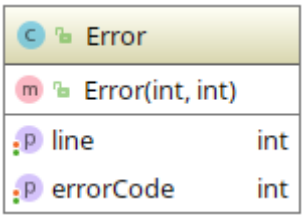


Clase Token

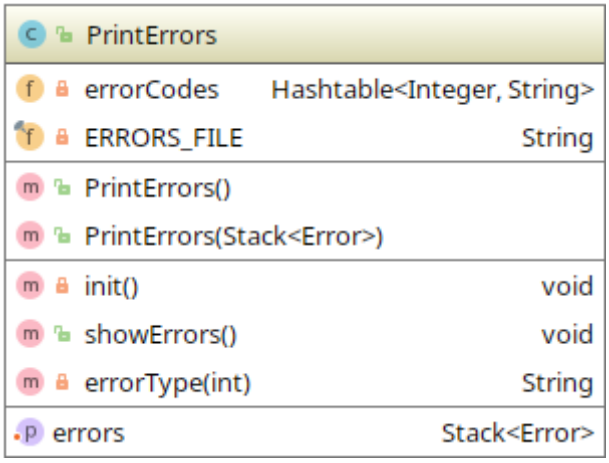


Package Exceptions

Clase Error

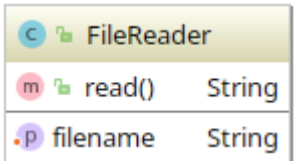


Clase PrintErrors



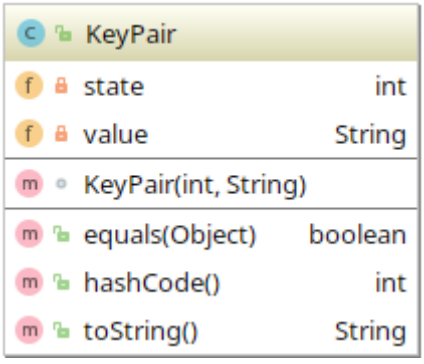
Package Files

Clase FileReader

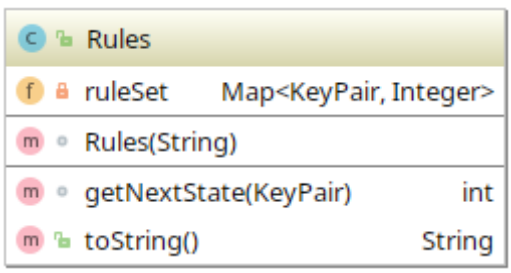


Package Automaton

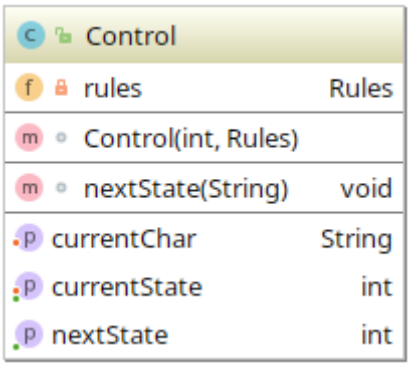
Clase KeyPair



Clase Rules



Clase Control



Clase Automaton

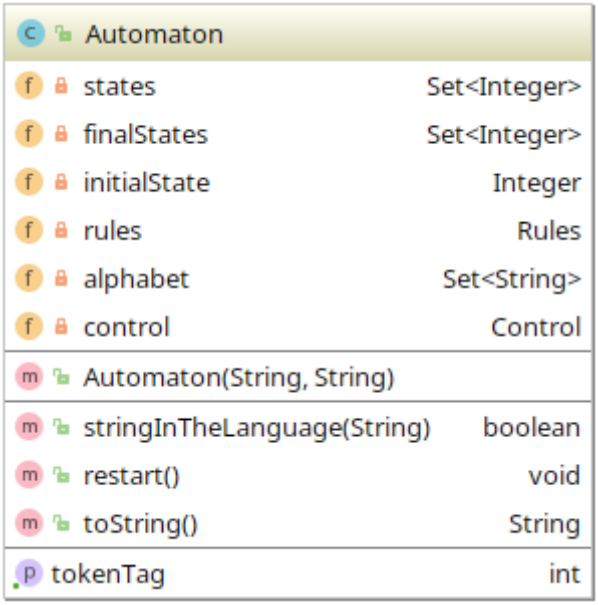


TABLA DE SÍMBOLOS, CON LA JUSTIFICACIÓN DE CADA UNA DE LAS COLUMNAS QUE LA INTEGREN.

Nombre columna	Tipo
id	String
tipo	String
línea de declaración	int

Se debe de almacenar el nombre del token que se encontró, que se guardará como **id**, el **tipo** (id, reserved); por último, la **línea de declaración**, para los análisis posteriores.

PILA DE ERRORES.

Se hará uso de una pila de enteros, donde se almacenarán los errores que se vayan generando.

Se hará uso de la clase `java.util.Stack` para evitar escribir código que resultaría innecesario e irrelevante para la construcción del analizador léxico.

PROCESOS DE APERTURA Y LECTURA DEL ARCHIVO DE CÓDIGO FUENTE, ASÍ COMO DEL PROCESO DE TOKENIZACIÓN.

Apertura y lectura del archivo de código fuente

Apertura

La apertura del archivo es básicamente una llamada a una función del sistema operativo, pero desde el lenguaje de programación únicamente se hace la llamada a la función que implementa la llamada al sistema operativo.

Lectura

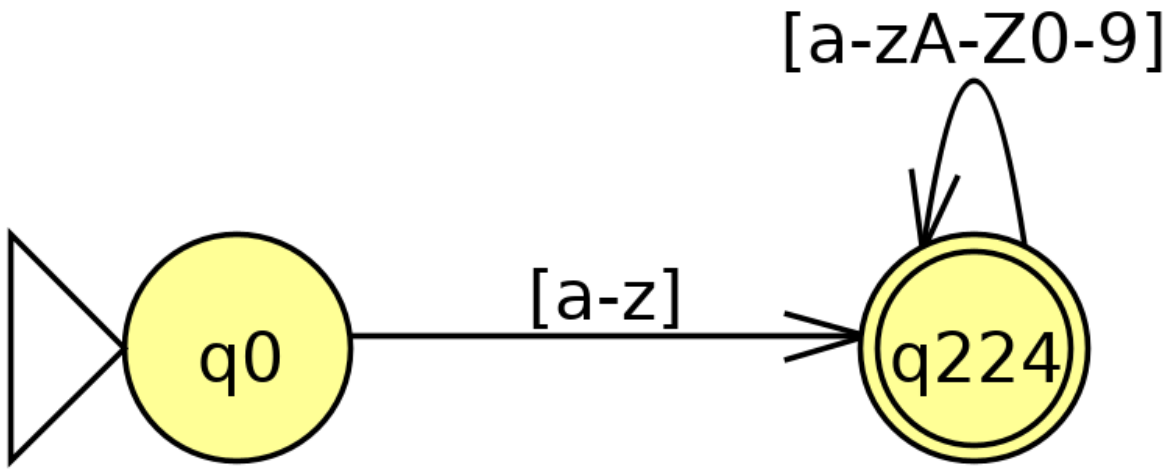
Se realiza la lectura línea a línea del archivo y las líneas leídas son almacenadas en un buffer hasta que se termina el archivo.

Tokenización

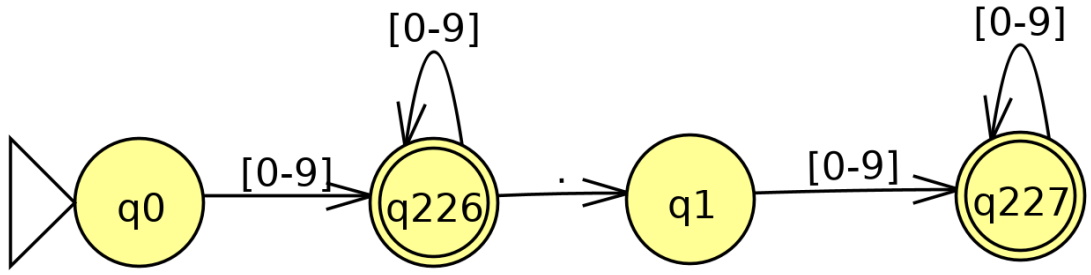
Se lee el código carácter a carácter hasta encontrar algún delimitador implícito (espacio, tabulador o salto de línea) o un cambio de contexto (paréntesis, punto y coma, operador relacional, operador de asignación, etc.), en ese momento, se toma el lexema y se ingresa en varios autómatas hasta que encuentre a que categoría pertenece, en caso que no pertenezca a ninguna categoría, se agrega el error a la pila de errores.

DISEÑO, PRUEBAS, MODELADO E IMPLEMENTACIÓN DE LOS AUTÓMATAS PARA LA CATEGORIZACIÓN DE CADA UNO DE LOS TOKENS GENERADOS.

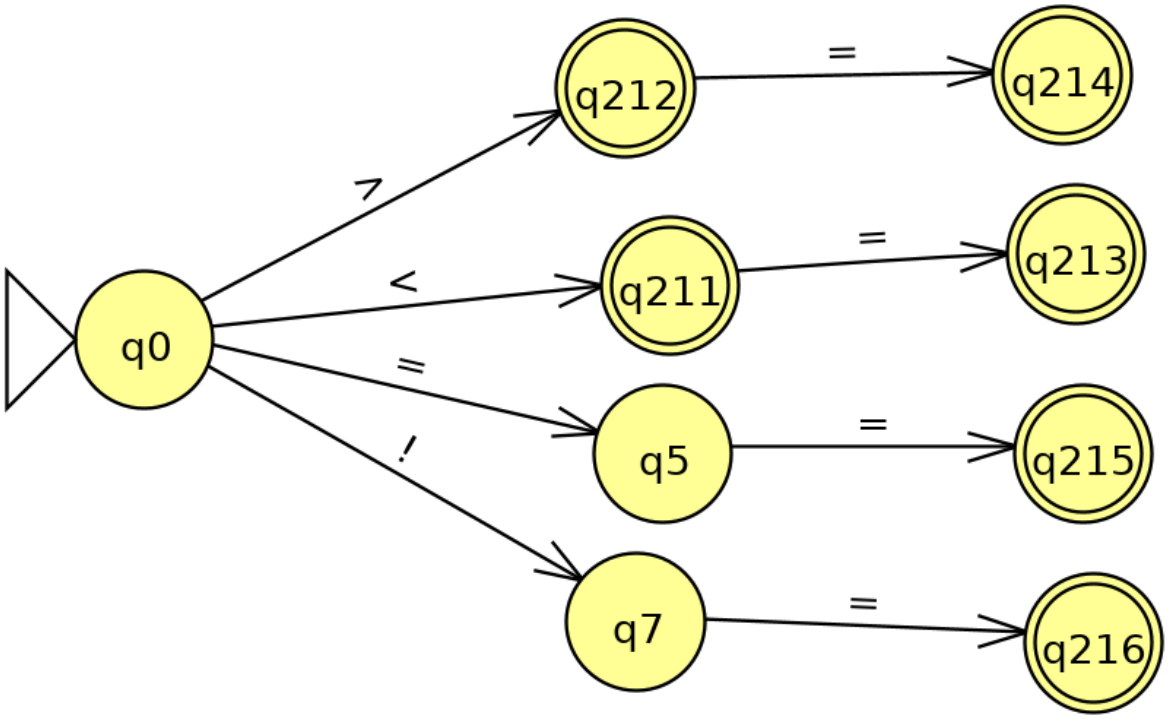
Identificadores



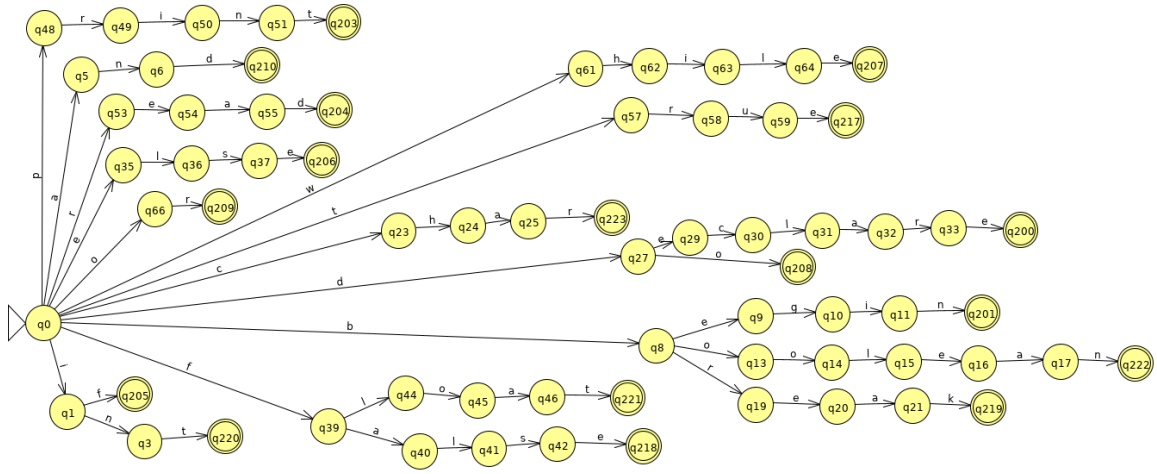
Números



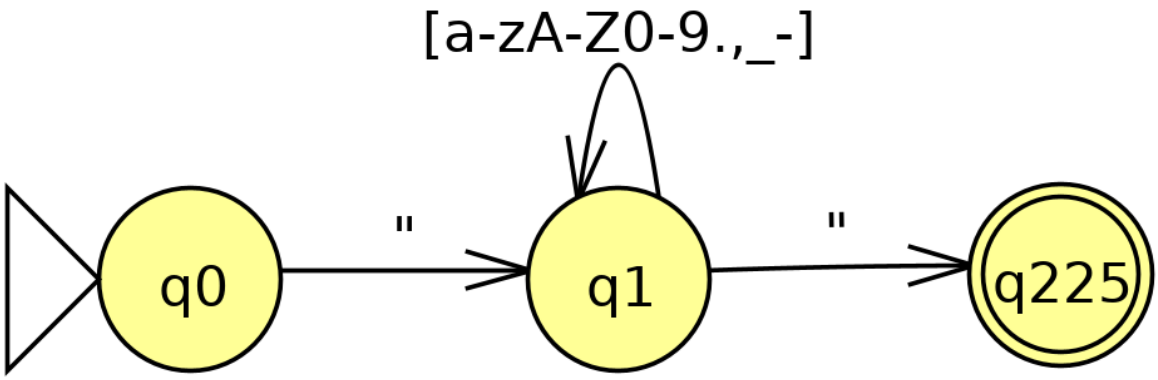
Operadores relacionales



Palabras reservadas



Cadenas



Elementos unarios

NOTA IMPORTANTE. A comparación del documento enviado el viernes 27 de septiembre, en este autómata, el estado 230, se separó en 4 estados distintos (230, 235, 236, 237), esto para tener un mejor control sobre cada operador, anteriormente para pasar al estado 230, se requería uno de los siguientes operadores: +,-,*, /. Ahora cada uno de ellos, representa un estado diferente.