

MANUAL TÉCNICO

IDE:

Netbeans 8.2

Version de JAVA:

java versión “1.8.0-201”

JAR agregados:

- java – cup – 11b.jar
- java – cup – 11b-runtime.jar
- jflex – full – 1.7.0.jar

jFlex:

Jflex es una herramienta desarrollada en Java que toma como entrada un archivo “entrada”, con este crea un archivo fuente java entrada.lex.java correspondiente al analizador léxico. En otras palabras, Jflex es un generador de analizadores léxicos. Los analizadores léxicos toman como entrada una cadena de caracteres y la convierten en una secuencia de tokens.

JCup:

JAVA CUP es un parser-generator. Es un analizador sintáctico que construye un parser para gramáticas tipo LALR(1), con código de producción y asociación de fragmentos de código JAVA. Cuando una producción en particular es reconocida, se genera un archivo fuente Java, parser.java que contiene una clase parser, con un método Symbol parser().

Instalacion de Jflex y Jcup:

1. JDK (Descarga el JDK de la página Web de Sun para los sistemas Operativos Linux, Windows y Solaris). Si se tiene instalado alguna versión de JAVA, solo es necesario buscar el directorio en el que se encuentra el JDK de esta.
2. Archivo para la generación de las clases para Jflex llamado Main.java que se puede descargar de la página Web de Jflex.
3. Descarga del sitio Web de CUP el código fuente de los archivos necesarios.

CLASES:

ClaseJava: Clase objeto que se encarga de guardar los objetos de las clases.

Constantes: Clase objeto que guarda los atributos de las constantes de C.

VerifObjetos: clase que se encarga de las verificaciones dentro de C.

OperacionesC: Se encarga de las operaciones que se lleven a cabo dentro del syntaxC

OperacionesJAVA: Se encarga de las operaciones que se lleven a cabo dentro del syntaxJava

OperacionesPY: Se encarga de las operaciones que se lleven a cabo dentro del syntaxPY

OperacionesVB: Se encarga de las operaciones que se lleven a cabo dentro del syntaxVB

TablaSimbolos: clase que guarda todos los objetos y atributos de todos los lenguajes.

Generacion: Se encarga de la generacion de codigo assembler

Objetos: Guarda los objetos del lenguaje Assembler

CreacionCodigo: Se encarga de la creacion del codigo 3 direcciones no ejecutable.

CreacionEjecutable: Se encarga de la creacion del codigo 3 direcciones ejecutable.

SyntaxC: Clase que guarda el analisis semántico, sintáctico y léxico del lenguaje C

SyntaxJAVA: Clase que guarda el análisis semántico, sintáctico y léxico del lenguaje Java

SyntaxPYTHON: Clase que guarda el análisis semántico, sintáctico y léxico del lenguaje Python

SyntaxVB: Clase que guarda el análisis semántico, sintáctico y léxico del lenguaje Visual Basic

guardado: Se encarga del guardado de los archivos asm y c

NumeracionLineas: A cargo de la numeracion de lineas dentro de los textArea.

ManejoTexto: Se encarga de la separacion de los codigos.

Panel3D: Panel con el codigo 3 direcciones no ejecutable.

Panel Assembler: Panel con el codigo assembler.

PanelEjecutable: Panel con el codigo 3 direcciones ejecutable.

Principal: Frame principal

ManejoC: Se encarga del manejo de los cuartetos pertenecientes a C.

ManejoJava: Se encarga del manejo de los cuartetos pertenecientes a Java.

ManejoPython: Se encarga del manejo de los cuartetos pertenecientes a Python.

ManejoVB: Se encarga del manejo de los cuartetos pertenecientes a Visual Basic.

ExeC: Clase encargada de la creacion de la tabla de simbolos de C.

ExeJava: Clase encargada de la creacion de la tabla de simbolos de Java.

ExePython: Clase encargada de la creacion de la tabla de simbolos de Python.

ExeVb: Clase encargada de la creacion de la tabla de simbolos de Visual Basic.

Arreglo: Se encarga de guardar los atributos de un arreglo.

Clase: Se encarga de guardar los atributos de una clase.

Metodo: Se encarga de guardar los atributos de un metodo.

ObjetosC: Se encarga de guardar los objetos del lenguaje C.

ObjetosJAVA: Se encarga de guardar los objetos del lenguaje Java.
ObjetosPYTHON: Se encarga de guardar los objetos del lenguaje Python.
ObjetosVB: Se encarga de guardar los objetos del lenguaje Visual Basic.
Parametro: Se encarga de guardar los atributos de un parametro.
Variable: Se encarga de guardar los atributos de una variable.

VerifC: Se encarga de las verificaciones semanticas de C.
VerifJava: Se encarga de las verificaciones semanticas de Java.
VerifPython: Se encarga de las verificaciones semanticas de Python.
VerifVb: Se encarga de las verificaciones semanticas de Vb.

GRAMÁTICA VB:

start with B

B ::= METODOS B

| COMENTARIO_A B
| COMENTARIO_B B
|
| error B

VARIABLES ::= Dim id C CONC_VAR_DEF

ASIGNACIONES ::= id igual VALOR CONC_VAR_AS
| condThis punto id igual VALOR CONC_VAR_AS

CONC_VAR_DEF ::= coma id C CONC_VAR_DEF
| puntoComa ENTER

CONC_VAR_AS ::= coma id C CONC_VAR_AS
| puntoComa ENTER

C ::= condAs condTipo
|

VAR ::= tipoInt
| tipoChar
| tipoFloat

OPERACION ::= parentesis_a OPERACION parentesis_b
| OPERACION mas OPERACION
| OPERACION menos OPERACION
| OPERACION por OPERACION
| OPERACION div OPERACION
| OPERACION modulo OPERACION
| TIPO_VAL

TIPO_VAL ::= id
| numero
| numero punto numero
| comillaSimple id comillaSimple
| comillaSimple numero comillaSimple
| menos id
| menos numero
| menos numero punto numero

METODOS ::= tipoPublic condSub id parentesis_a PARAMETROS parentesis_b
 CUERPO_METODO condFin condSub

| tipoPublic condFunction id parentesis_a PARAMETROS parentesis_b condAs condTipo
CUERPO_METODO condFin condSub

PARAMETROS ::= PARAMETROS2 id condAs condTipo

PARAMETROS2 ::= PARAMETROS2 condAs condTipo id coma

TIPO_METODO ::= tipoChar]

| tipoFloat

| tipoChar

| tipoVoid

CUERPO_METODO ::= CUERPO_METODO VARIABLES

| CUERPO_METODO ASIGNACIONES

| CUERPO_METODO IF

| CUERPO_METODO FOR

| CUERPO_METODO WHILE

| CUERPO_METODO DO_WHILE

| CUERPO_METODO MENSAJE

| CUERPO_METODO SELECT

| CUERPO_METODO LECTURA_DATOS

| CUERPO_METODO COMENTARIO_A

| CUERPO_METODO COMENTARIO_B

| CUERPO_METODO error

RETORNO ::= retorno OPERACION puntoComa ENTER

IF ::= condIf EXT2 condThen CUERPO_METODO ELSE condFin condif

ELSE ::= condElseIf EXT2 condThen CUERPO_METODO ELSE

| condElse condThen CUERPO_METODO RETORNO

| ENTER

FOR ::= codFor parentesis_a INT id igual VAL_FOR puntoComa EXT2 puntoComa id INC
parentesis_b llave_a CUERPO_METODO llave_b

INT ::= tipoInt

INC ::= igual OPERACION

OP ::= menor igual

| menor

| mayor igual

| mayor

| igual
| igual igual
| admiracion igual

WHILE ::= condWhile EXT2 CUERPO_METODO condEndWhile

DO_WHILE ::= condDo CUERPO_METODO condLoop condWhile EXT2

SELECT ::= condSelect conCase parentesis_a id parentesis_b CASOS condEnd condSelect

CASOS ::= condCase VAL_SELECT CUERPO_METODO CASOS

| condCase condElse CUERPO_METODO

|

EXT2 ::= EXT2 and EXT2

| EXT2 or EXT2

| BOOLEAN3

| admiracion parentesis_a BOOLEAN3 parentesis_b

| admiracion BOOLEAN3

BOOLEAN3 ::= OPERACION OP COPERACION

LECTURA_DATOS ::= id igual TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa

| TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa

TIPO_INPUT ::= intInput

| FloatInput

| CharInput

TEXTO ::= VAL TEXTO2

| comilla TEXTO_PLANO comilla TEXT2

TEXTO2 ::= mas TEXTO

| ampersand TEXTO

|

COMENTARIO_A ::= comentario_a TEXTO_PLANO

COMENTARIO_B ::= comentario_b TEXTO_PLANO Comentario_c

GRAMATICA JAVA

start with AA

A ::= A tipoPublic clase id llaveA ENTER B llaveB ENTER

B ::= VARIABLES B
| METODOS B
| CONSTRUCTOR B
| COMENTARIO_A B
| COMENTARIO_B B
|
| error B

VARIABLES ::= VAR id C CONC_VAR_DEF

ASIGNACIONES ::= id igual VALOR CONC_VAR_AS
| condThis punto id igual VALOR CONC_VAR_AS

CONC_VAR_DEF ::= coma id C CONC_VAR_DEF
| puntoComa ENTER

CONC_VAR_AS ::= coma id C CONC_VAR_AS
| puntoComa ENTER

C ::= igual VALOR
|

VAR ::= tipoInt
| tipoChar
| tipoFloat

OPERACION ::= parentesis_a OPERACION parentesis_b
| OPERACION mas OPERACION
| OPERACION menos OPERACION
| OPERACION por OPERACION
| OPERACION div OPERACION
| OPERACION modulo OPERACION
| TIPO_VAL

TIPO_VAL ::= id
| numero
| numero punto numero
| comillaSimple id comillaSimple
| comillaSimple numero comillaSimple
| menos id
| menos numero
| menos numero punto numero

METODOS ::= tipoPublic TIPO_METODO id parentesis_a PARAMETROS parentesis_b llave_a
CUERPO_METODO RETORNO llave_b ENTER
| tipoPublic TIPO_METODO id parentesis_a parentesis_b llave_a CUERPO_METODO
RETORNO llave_b ENTER

CONSTRUCTOR ::= tipoPublic id parentesis_a PARAMETROS parentesis_b llave_a
CUERPO_METODO llave_b ENTER
| tipoPublic parentesis_a parentesis_b llave_a CUERPO_METODO llave_b ENTER

PARAMETROS ::= PARAMETROS2 VAR id

PARAMETROS2 ::= PARAMETROS2 VAR id coma
|

TIPO_METODO ::= tipoChar]
| tipoFloat
| tipoChar
| tipoVoid

CUERPO_METODO ::= CUERPO_METODO VARIABLES
| CUERPO_METODO ASIGNACIONES
| CUERPO_METODO IF
| CUERPO_METODO FOR
| CUERPO_METODO WHILE
| CUERPO_METODO DO_WHILE
| CUERPO_METODO MENSAJE
| CUERPO_METODO SWITCH
| CUERPO_METODO LECTURA_DATOS
| CUERPO_METODO COMENTARIO_A
| CUERPO_METODO COMENTARIO_B
| CUERPO_METODO error

RETORNO ::= retorno OPERACION puntoComa ENTER
|

IF ::= condIf parentesis_a EXT2 parentesis_b llave_a CUERPO_METODO RETORNO llave_b

ELSE ::= condElse condIf parentesis_a EXT2 parentesis_b llave_a CUERPO_METODO RETORNO
llave_b
| condElse llave_a CUERPO_METODO RETORNO llave_b
| ENTER

FOR ::= codFor parentesis_a INT id igual VAL_FOR puntoComa EXT2 puntoComa id INC
parentesis_b llave_a CUERPO_METODO llave_b

INT ::= tipoInt
|

INC ::= igual OPERACION

OP ::= menor igual
| menor
| mayor igual
| mayor
| igual
| igual igual
| admiracion igual

WHILE ::= condWhile parentesis_a EXT2 parentesis_b llave_a CUERPO_METODO llave_b

DO_WHILE ::= condDo llave_a CUERPO_METODO condWhile parentesis_a EXT2 parentesis_b
puntoComa

SWITCH ::= condSwitch parentesis_a id parentesis_b llave_a CASOS llave_b

CASOS ::= noCase VAL_SWITCH dosPuntos CUERPO_METODO BREAK CASOS
| default dosPuntos CUERPO_METODO BREAK
|

BREAK ::= quiebre puntoComa
| RETORNO

EXT2 ::= EXT2 and EXT2
| EXT2 or EXT2
| BOOLEAN3
| admiracion parentesis_a BOOLEAN3 parentesis_b
| admiracion BOOLEAN3

BOOLEAN3 ::= OPERACION OP COPERACION

LECTURA_DATOS ::= id igual TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa
| TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa

TIPO_INPUT ::= intInput
| FloatInput
| CharInput

TEXTO ::= VAL TEXTO2
| comilla TEXTO_PLANO comilla TEXT2

TEXTO2 ::= mas TEXTO
|

COMENTARIO_A ::= comentario_a TEXTO_PLANO

COMENTARIO_B ::= comentario_b TEXTO_PLANO Comentario_c

GRAMATICA PYTHON:

INICIO1 ::= ENTER2 INICIO

;

INICIO ::= INICIO METODOS

1

	error
--	-------

METODOS ::= condDef id parenthesis_a PARAMETROS parenthesis_b dosPuntos ENTER
CUERPO_METODO

$$\text{DEF} ::= \text{condDef}$$

1

| error

METODO_COND ::= id parentesis_a PARAM_METODOS parentesis_b

METODO_VOID ::= TAB id parenthesis_a PARAM_METODOS parenthesis_b ENTER

$$\text{PARAM_METODOS} ::= \text{PARAM}$$

1

$$\text{PARAM} ::= \text{PARAM2 OPERACION}$$

1

| error

PARAM2 ::= PARAM2 OPERACION coma

1

error

$$\text{PARAMETROS} ::= \text{id PARAMETROS2}$$

1

| error

RETURN ::= TAB condReturn RETORNO ENTER

RETORNO ::= OPERACION

1

	error
--	-------

$$\text{OPERACION} ::= \text{parentesis_a OPERACION parentesis_b}$$

1

OPERACION mas OPERACION

1

OPERACION menos OPERACION

1

OPERACION por OPERACION

1

OPERACION div OPERACION

1

OPERACION	porcentaje	OPERACION
1	100	1
2	100	2
3	100	3
4	100	4
5	100	5
6	100	6
7	100	7
8	100	8
9	100	9
10	100	10
11	100	11
12	100	12
13	100	13
14	100	14
15	100	15
16	100	16
17	100	17
18	100	18
19	100	19
20	100	20
21	100	21
22	100	22
23	100	23
24	100	24
25	100	25
26	100	26
27	100	27
28	100	28
29	100	29
30	100	30
31	100	31
32	100	32
33	100	33
34	100	34
35	100	35
36	100	36
37	100	37
38	100	38
39	100	39
40	100	40
41	100	41
42	100	42
43	100	43
44	100	44
45	100	45
46	100	46
47	100	47
48	100	48
49	100	49
50	100	50
51	100	51
52	100	52
53	100	53
54	100	54
55	100	55
56	100	56
57	100	57
58	100	58
59	100	59
60	100	60
61	100	61
62	100	62
63	100	63
64	100	64
65	100	65
66	100	66
67	100	67
68	100	68
69	100	69
70	100	70
71	100	71
72	100	72
73	100	73
74	100	74
75	100	75
76	100	76
77	100	77
78	100	78
79	100	79
80	100	80
81	100	81
82	100	82
83	100	83
84	100	84
85	100	85
86	100	86
87	100	87
88	100	88
89	100	89
90	100	90
91	100	91
92	100	92
93	100	93
94	100	94
95	100	95
96	100	96
97	100	97
98	100	98
99	100	99
100	100	100

1

TIPO_VAL

$$\text{TIPO_VAL} ::= \text{id}$$

1

	numero
--	--------

1

METODO_COND

```

| menos METODO_COND
| numero punto numero
| comillaSimple id comillaSimple
| comillaSimple numero comillaSimple
| menos comillaSimple id comillaSimple
| menos comillaSimple numero comillaSimple
| menos id
| menos numero
| menos numero punto numero

```

TIPO_VAL2 ::= comilla id comilla

```

| id
| numero
| numero punto numero

```

CUERPO_METODO ::= VARIABLES3 CUERPO_METODO

```

| FOR CUERPO_METODO
| WHILE CUERPO_METODO
| IF CUERPO_METODO
| BORRAR CUERPO_METODO
| MENSAJE CUERPO_METODO
| LECTURA_DATOS CUERPO_METODO
| ELSE CUERPO_METODO
| condPass ENTER CUERPO_METODO
| COMENTARIO_B CUERPO_METODO
| COMENTARIO_A CUERPO_METODO
| RETURN CUERPO_METODO
| METODO_VOID CUERPO_METODO

```

VARIABLES3 ::= TAB id igual OPERACION VARIABLES2

```

| TAB LADO1 igual LADO2
| TAB id igual error VARIABLES2

```

VARIABLES2 ::= coma id igual OPERACION VARIABLES2

```

| igual OPERACION VARIABLES2 ENTER
| coma id igual error VARIABLES2
| igual error VARIABLES2

```

LADO1 ::= id

```

| coma id
| VARS2

```

VARS2 ::= coma id VARS2

LADO2 ::= OPERACION coma OPERACION VARS4

VARS4 ::= coma OPERACION VARS4

```

| ENTER

```

| coma error VARS4

FOR ::= TAB condFor id condIn condRange parentesis_a RANGE parentesis_b dosPuntos ENTER

RANGE ::= OPERACION

| OPERACION coma OPERACION
| OPERACION coma OPERACION coma OPERACION
| OPERACION coma error
| error

OP_FOR ::= id

| menos id
| numero
| menos numero
| error

WHILE ::= TAB condWhile EXT dosPuntos ENTER

EXT ::= EXT and EXT

| EXT or EXT
| EXT
| CONDICION
| not parentesis_a CONDICION parentesis_b
| not CONDICION
| error

CONDICION ::= OPERACION OP OPERACION

OP ::= igual igual

| mayor
| menor
| mayorIgual
| menorIgual
| admiracionIgual
| error

IF ::= TAB condElse dosPuntos ENTER

| TAB condElif EXT dosPuntos ENTER

ELSE ::= TAB condElse dosPuntos ENTER

| TAB condElif EXT dosPuntos ENTER

BORRAR ::= TAB condDel id ENTER

LECTURA_DATOS ::= TAB id igual TIPO_INPUT parentesis_a TEXTO parentesis_b ENTER

| TAB id igual TIPO_INPUT parentesis_a parentesis_b ENTER
| TAB TIPO_INPUT parentesis_a TEXTO parentesis_b ENTER
| TAB TIPO_INPUT parentesis_a parentesis_b ENTER

TIPO_INPUT ::= intInput
| charInput
| floataInput

MENSAJES ::= TAB mensaje parentesis_a TEXTO parentesis_b ENTER

CONC ::= coma
| mas
| error

TEXTO ::= VAR_MENSAJE RESTO_MENSAJE comilla TEXTO2 comilla RETO_MENSAJE

RESTO_MENSAJE ::= CONC TEXTO
|

ID ::= coma id ID
|

VAR_MENSAJE ::= id
| numero
| numero punto numero
| menos id
| menos numero
| menos numero punto numero

ENTER ::= ENTER enter
| enter

ENTER2 ::= ENTER enter
| enter

TAB ::= TAB tab
| tab

COMENTARIO_A ::= comentario_a TEXTO3 ENTER

COMENTARIO_B ::= comentario_b TEXTO2 comentario_c