

GRAMÁTICAS

CONJ = "CONJ"

- DOSPUNTOS = ":"
- COMA = ","
- TILDE = "~"
- LLAVEA = "{"
- LLAVEC = "}"
- CONCATENACION = "."
- DISYUNCION = "|"
- ASTERISCO = "*"
- SUMA = "+"
- INTERROGACION = "?"
- PORCENTAJE = "%"
- PUNTO_Y_COMA = ";
- COMILLAS = "\"
- ENTER = [\ \n]
- SPACE = [\ \t\r\f]
- ESCAPADOS = "\\n" | "\\\" | "\\\""
- NO_ESCAPADOS = [^\"]
- COMENTARIOMULTI = "<!" [^]~ ">"
- COMENTARIOSIMPLE = "//" . *
- MINUSCULAS = [a-z]
- MAYUSCULAS = [A-Z]
- NUMEROS = [0-9]
- CARACTER = (\ {NO_ESCAPADOS} \) | {ESCAPADOS}
- IDENTIFICADOR = [a-zA-Z0-9]+
- CADENA = \" ([^\"] | "\\\"")+\"
- CARACTERES = [!-\\:-=?-@\\[-`{-}]
- FLECHA = "-" {SPACE}* ">"

TERMINALES

CONJ, DOSPUNTOS, FLECHA, COMA, TILDE, LLAVEA, LLAVEC, CONCATENACION, DISYUNCION, ASTERISCO, SUMA, INTERROGACION, PORCENTAJE, PUNTO_Y_COMA, IDENTIFICADOR, MINUSCULAS, MAYUSCULAS, NUMEROS, CARACTERES, CARACTER, CADENA;

NO TERMINALES

conjuntos, comprobacion, instruccion, regex, notacion,
agrupacion, singular, evaluacion;

PRODUCCIONES

codigo::= LLAVEA conjuntos PORCENTAJE PORCENTAJE
PORCENTAJE PORCENTAJE comprobacion LLAVEC;

conjuntos::= conjuntos instruccion |
instruccion;

instruccion::= CONJ DOSPUNTOS IDENTIFICADOR:hola
FLECHA notacion:b PUNTO_Y_COMA {: conjuntos.add(new
Conjunto(hola,(ArrayList) b)); :}
IDENTIFICADOR:hola FLECHA regex:a
PUNTO_Y_COMA {:
arboles.add(new AFD(hola, (arbol)a));
:};

notacion::= MAYUSCULAS:a TILDE MAYUSCULAS:b {:
ArrayList<Integer> nuevo_conjunto = new
ArrayList<Integer>();
for(int i = (int) a.charAt(0);i<=(int) b.charAt(0);i++){
nuevo_conjunto.add(i);
}RESULT = nuevo_conjunto;
:}
MINUSCULAS:a TILDE MINUSCULAS:b {:
ArrayList<Integer> nuevo_conjunto = new
ArrayList<Integer>();
for(int i = (int) a.charAt(0);i<=(int) b.charAt(0);i++){
nuevo_conjunto.add(i);
}RESULT = nuevo_conjunto;
:}
NUMEROS:a TILDE NUMEROS:b {:
ArrayList<Integer> nuevo_conjunto = new
ArrayList<Integer>();
for(int i = (int) a.charAt(0);i<=(int) b.charAt(0);i++){
nuevo_conjunto.add(i);
}RESULT = nuevo_conjunto;
:}
CARACTERES:a TILDE CARACTERES:b {:

```

        ArrayList<Integer>    nuevo_conjunto    =    new
ArrayList<Integer>();
        for(int i = (int) a.charAt(0);i<=(int) b.charAt(0);i++){
            nuevo_conjunto.add(i);
        }RESULT = nuevo_conjunto;
    :}
    agrupacion:a {:
        String notacion = (String) a;
        ArrayList<Integer>    nuevo_conjunto    =    new
ArrayList<>();
        for(int i = 0;i < notacion.length();i++)
        {
            nuevo_conjunto.add((int) notacion.charAt(i));
        }
        RESULT = nuevo_conjunto;
    :};

```

```

agrupacion::= agrupacion:a COMA singular:b {:RESULT =
(String)a+(String)b;:}|
    singular:a {:RESULT = a;:};

```

```

singular::= MAYUSCULAS:a {:RESULT = a;:} |
    MINUSCULAS:a {:RESULT = a;:} |
    CARACTERES:a {:RESULT = a;:} |
    NUMEROS:a {:RESULT = a;:};

```

```

regex::= CONCATENACION:a regex:b regex:c {:
    arbol padre = new arbol(a);
    padre.setHijol((arbol)b);
    padre.setHijoD((arbol)c);
    RESULT = padre;
:} |
DISYUNCION:a regex:b regex:c {:
    arbol padre = new arbol(a);
    padre.setHijol((arbol)b);
    padre.setHijoD((arbol)c);
    RESULT = padre;
:} |
ASTERISCO:a regex:b {:
    arbol padre = new arbol(a);
    padre.setHijol((arbol)b);
    RESULT = padre;
:} |
SUMA:a regex:b {:

```

```

        arbol padre = new arbol(a);
        padre.setHijol((arbol)b);
        RESULT = padre;
    :} |
INTERROGACION:a regex:b {:
    arbol padre = new arbol(a);
    padre.setHijol((arbol)b);
    RESULT = padre;
    :} |
CARACTER:a {:
    arbol hoja=new arbol(a);
    hoja.setHoja(true);
    hoja.setAnulable(false);
    RESULT = hoja;
    :} |
LLAVEA IDENTIFICADOR:a LLAVEC {:
    arbol hoja=new arbol(a);
    hoja.setHoja(true);
    hoja.setAnulable(false);
    RESULT = hoja;
    :};

```

```

comprobacion::= comprobacion evaluacion |
                evaluacion;

```

```

evaluacion::= IDENTIFICADOR:a DOSPUNTOS CADENA:b
PUNTO_Y_COMA {: pruebas.add(new Evaluacion(a,b));;}|
                IDENTIFICADOR:a DOSPUNTOS CARACTER:b
PUNTO_Y_COMA {: pruebas.add(new Evaluacion(a,b));;};

```