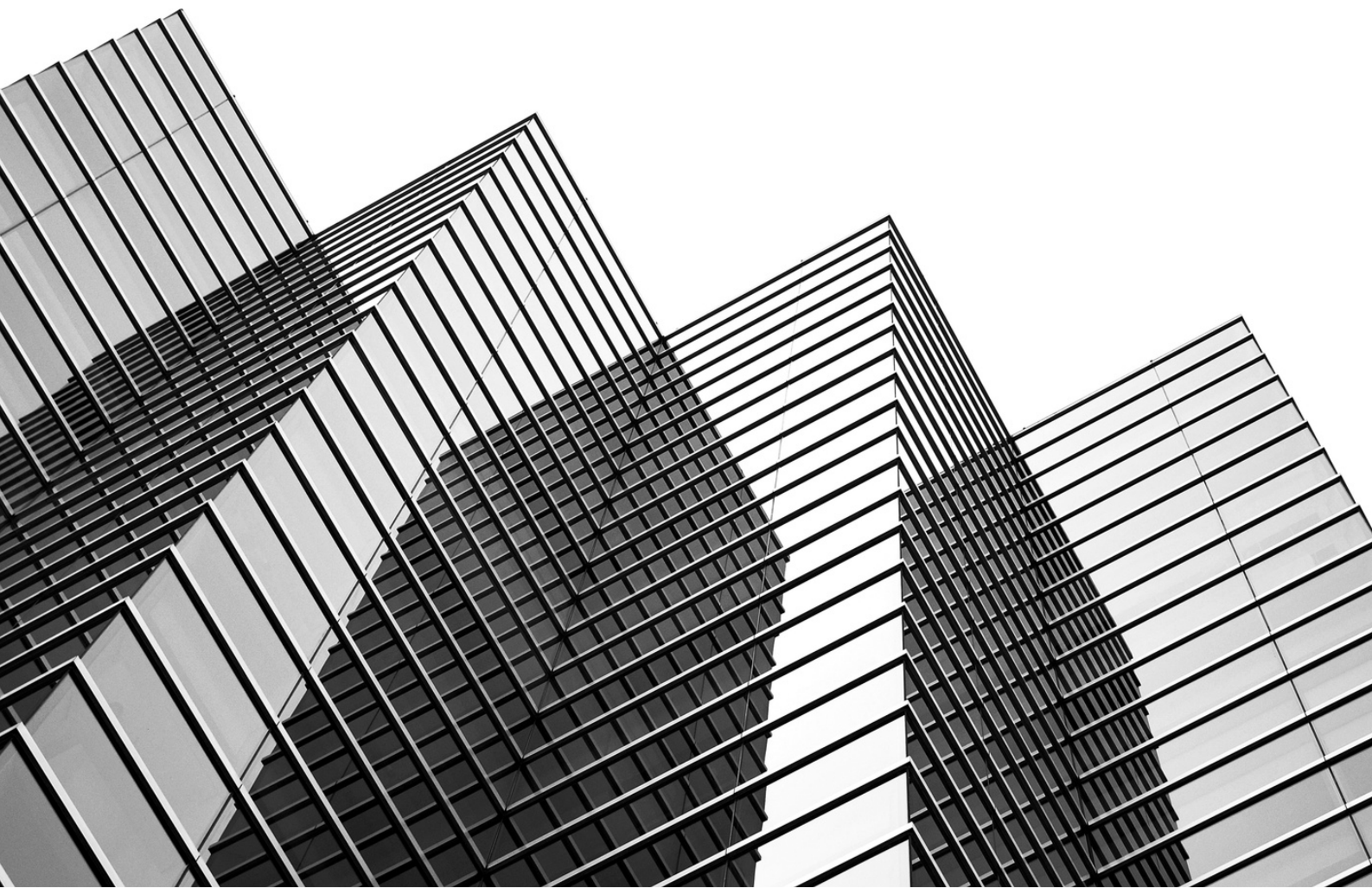




Compiladores y  
Ensambladores 1

# GRAMÁTICA



Joshua David Osorio Tally  
202110773

# Expresiones

01

```
tkRconj = "CONJ"  
tkDosPuntos=":"  
tkFlecha = "-" {Espacio}* ">"  
tkComa = ","  
tkOnda="~"  
tkLlaveA="{"  
tkLlaveC="}"  
tkConcatenacion="."  
tkDisyuncion="|"  
tkKleene="*"
```

# 02

```
tkPositiva="+"
tkBooleana="?"
tkPorcentaje="%"
tkPuntoComa=";"
Espacio= [ \t\r\n]+
CMultilinea="<!" ([^!] | ("!" [^\>]))+ "!">"
CSimple="//":
```

03

```
tkMinuscula= [a-z]
tkMayuscula= [A-Z]
tkNumero = [0-9]
tkEspeciales = [-/:@\[-\{-}]
Escapados = "\\\"\\'\\\\\\|\\\\\\|\\\\\\|\\\\\\|\\\\\\n"
NoEscapados = [^\\"]
tkIdentificador = [a-zA-Z_][a-zA-Z0-9_]+
tkChar = ("{"NoEscapados}"|{"Escapados"})
tkString = "\"([^\"]|\"\\\\\\\\\")+\""
```

# Terminales



## No Terminales

1. tkRconj
2. tkDosPuntos
3. tkFlecha
4. tkComa
5. tkOnda
6. tkLlaveA
7. tkLlaveC
8. tkConcatenacion
9. tkDisyuncion
10. tkKleene
11. tkPositiva
12. tkBooleana
13. tkPorcentaje
14. tkPuntoComa
15. tkMinuscula
16. tkMayuscula
17. tkNumero
18. tkEspeciales
19. tkIdentificador
20. tkChar
21. tkString

1. Programa
2. declaraciones
3. declaracion
4. notacion\_conjuntos
5. contenedor\_conj\_comas
6. conj\_comas
7. expresion\_regular
8. evaluaciones
9. evaluacion

# Producciones

01

1. Programa ::= tkLlaveA declaraciones tkPorcentaje  
tkPorcentaje tkPorcentaje tkPorcentaje evaluaciones  
tkLlaveC;

02

2. declaraciones ::= declaraciones declaracion |  
declaracion;

03

3. declaracion ::= tkRconj tkDosPuntos tkIdentificador:a  
tkFlecha notacion\_conjuntos:b tkPuntoComa {:  
conjuntos.add(new Conjunto(a,(ArrayList) b)); :}  
tkIdentificador:a tkFlecha expresion\_regular:b  
tkPuntoComa {: expresiones.add(new Expresion\_regular(a,  
(Nodo)b)); :};

04

4. expresion\_regular ::= tkConcatenacion:a  
expresion\_regular:b expresion\_regular:c {:  
Nodo nuevo = new Nodo(a);  
nuevo.setIzq((Nodo)b);  
nuevo.setDer((Nodo)c);  
RESULT = nuevo; :}  
tkDisyuncion:a expresion\_regular:b  
expresion\_regular:c {:  
Nodo nuevo = new Nodo(a);  
nuevo.setIzq((Nodo)b);  
nuevo.setDer((Nodo)c);  
RESULT = nuevo; :}  
tkKleene:a expresion\_regular:b {:  
Nodo nuevo = new Nodo(a);  
nuevo.setIzq((Nodo)b);  
RESULT = nuevo; :}  
tkPositiva:a expresion\_regular:b {:  
Nodo nuevo = new Nodo(a);  
nuevo.setIzq((Nodo)b);  
RESULT = nuevo; :}  
tkBooleana:a expresion\_regular:b {:  
Nodo nuevo = new Nodo(a);  
nuevo.setIzq((Nodo)b);  
RESULT = nuevo; :}  
tkLlaveA tkIdentificador:a tkLlaveC {: Nodo  
nuevo = new Nodo(a);  
nuevo.setHoja(true);  
RESULT = nuevo; :}  
tkChar:a {: Nodo nuevo = new Nodo(a);  
nuevo.setHoja(true);  
RESULT = nuevo; :};

# Producciones

05

```
5. notacion_conjuntos ::= tkMinuscula:a tkOnda
tkMinuscula: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;
    :}|
    tkMayuscula:a tkOnda tkMayuscula: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;
    :}|
    tkNumero:a tkOnda tkNumero: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;
    :}|
    tkEspeciales:a tkOnda tkEspeciales:b {
    ArrayList<Integer> nuevo_conjunto = new
ArrayList<Integer>();
    for(int i = (int) a.charAt(0); i <= (int) b.charAt(0); i++){
        if( (i > 47 && i < 58) || (i > 64 && i < 91) || (i > 96 &&
i < 123)){
            continue;
        }else{
            nuevo_conjunto.add(i);
        }
    } RESULT = nuevo_conjunto;
    :}|
    contenedor_conj_comas: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;
    :};
```



# Producciones

06

```
contenedor_conj_comas ::= contenedor_conj_comas:a  
tkComa conj_comas:b { :RESULT = (String)a+(String)b;: } |  
conj_comas:a { :RESULT = a;: };
```

07

```
conj_comas ::= tkMinuscula:a { :RESULT = a;: } |  
tkMayuscula:a { :RESULT = a;: } |  
tkNumero:a { :RESULT = a;: } |  
tkEspeciales:a { :RESULT = a;: };
```

08

```
evaluaciones ::= evaluaciones evaluacion |  
evaluacion;
```

09

```
evaluacion ::= tkIdentificador:a tkDosPuntos tkString:b  
tkPuntoComa { : pruebas.add(new Evaluacion(a,b));: } |  
tkIdentificador:a tkDosPuntos tkChar:b tkPuntoComa { :  
pruebas.add(new Evaluacion(a,b));: };
```