

# GRAMÁTICA

Pablo Josué Barahona Luncey

202109715

3560855890101

Organización de Lenguajes y Compiladores 1

# PROYECTO 1

## EXPRESIONES

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}\* ">"

# PROYECTO 1

## GRAMÁTICA

### TERMINALES

- 1.CONJ
- 2.DOSPUNTOS
- 3.FLECHA
- 4.COMA
- 5.TILDE
- 6.LLAVEA
- 7.LLAVEC
- 8.CONCATENACION
- 9.DISYUNCION
- 10.ASTERISCO
- 11.SUMA
- 12.INTERROGACION
- 13.PORCENTAJE
- 14.PUNTO\_Y\_COMA
- 15.IDENTIFICADOR
- 16.MINUSCULAS
- 17.MAYUSCULAS
- 18.NUMEROS
- 19.CARACTERES
- 20.CARACTER
- 21.CADENA

### NO TERMINALES

- 1.codigo
- 2.conjuntos
- 3.comprobacion
- 4.instruccion
- 5.regex
- 6.notacion
- 7.agrupacion
- 8.singular
- 9.evaluacion

# PROYECTO 1

## GRAMÁTICA

### PRODUCCIONES

1. `codigo::= LLAVEA conjuntos PORCENTAJE PORCENTAJE  
PORCENTAJE PORCENTAJE comprobacion LLAVEC;`
2. `conjuntos::= conjuntos instruccion |  
instruccion;`
3. `instruccion::= CONJ DOSPUNTOS IDENTIFICADOR:hola FLECHA  
notacion PUNTO_Y_COMA |  
IDENTIFICADOR:hola FLECHA regex:a PUNTO_Y_COMA {  
arboles.add(new AFD((arbol)a));  
};`
4. `notacion::= MAYUSCULAS TILDE MAYUSCULAS |  
MINUSCULAS TILDE MINUSCULAS |  
NUMEROS TILDE NUMEROS |  
CARACTERES TILDE CARACTERES |  
agrupacion;`
5. `agrupacion::= agrupacion COMA singular |  
singular;`
6. `singular::= MAYUSCULAS |  
MINUSCULAS |  
CARACTERES |  
NUMEROS;`
7. `comprobacion::= comprobacion evaluacion |  
evaluacion;`
8. `evaluacion::= IDENTIFICADOR DOSPUNTOS CADENA  
PUNTO_Y_COMA;`

# PROYECTO 1

## GRAMÁTICA

### PRODUCCIONES

9. 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;
```

# PROYECTO 1

## GRAMÁTICA

### PRODUCCIONES

```
    :} |  
    CHARACTER: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;  
    :};
```