

Procesadores del Lenguaje

Práctica final, quinta entrega

Autores: Álvaro Marco Pérez (100383382), David Rico Menéndez (100384036)

Datos: Grupo de trabajo 2

Índice

1. Estudio del código	2
2. Gramática realizada	3
3. Conclusiones	5

1. Pasos anteriores

Se ha realizado un estudio previo de la gramática con el objetivo de comprender el alcance inicial de la gramática.

La gramática original del fichero es:

2. Gramática ampliada

Para solventar las reglas correspondientes a los puntos 18 y 19 se ha realizado la siguiente gramática:

```
axioma:      globales funciones main
            ;
globales:    /* Lambda */
            |  declaracion_global ';' r_globales
            ;
r_globales:  /* Lambda */
            |  declaracion_global ';' r_globales
            ;
funciones:   /* Lambda */
            |  IDENTIF
              '(' args ')' '{' body '}' r_funciones
            ;
r_funciones: /* Lambda */
            |  IDENTIF
              '(' args ')' '{' body '}' r_funciones
            ;
main:        MAIN '(' args ')' '{' body '}'
            ;
args:        /* Lambda */
            |  INTEGER IDENTIF r_args
            ;
r_args:      /* Lambda */
            |  ';' INTEGER IDENTIF r_args
            ;
body:        sentencia ';' r_body
            |  WHILE '(' expresion ')' '{' body '}' r_body
            |  IF '(' expresion ')' '{' body '}' r_body
            |  IF '(' expresion ')' '{' body '}' ELSE '{' body '}' r_body
```

```

    | FOR '(' declaracion ';' expresion ';' incremento ')' '{' body '}' r_body
    | RETURN expresion r_expresion ';' r_body
    | expresion ';' r_body
    ;
r_body:      /* Lambda */
    | body
    ;
sentencia:   declaracion
    | PRINTF '(' STRING ',' expresion r_sentencia ')'
    | PUTS '(' STRING ')'
    ;
r_sentencia: /* Lambda */
    | ',' expresion r_sentencia
    ;
declaracion: INTEGER IDENTIF '=' expresion r_declaracion
    | INTEGER IDENTIF r_declaracion
    | IDENTIF '=' expresion
    | INTEGER IDENTIF '[' expresion ']' r_declaracion
    | IDENTIF '[' expresion ']' '=' expresion r_declaracion
    | IDENTIF ',' r_identif '=' expresion r_expresion
    ;
r_identif:   IDENTIF
    | IDENTIF ',' r_identif
declaracion_global:
    INTEGER IDENTIF '=' NUMBER r_declaracion
    | INTEGER IDENTIF r_declaracion
    | INTEGER IDENTIF '[' expresion ']' r_declaracion
    ;
r_declaracion: /* Lambda */
    | ',' r_declaracion
    | IDENTIF '=' expresion r_declaracion
    | IDENTIF r_declaracion
    | IDENTIF '[' expresion ']' r_declaracion
    ;
expresion:   termino
    | expresion '+' expresion
    | expresion '-' expresion
    | expresion '*' expresion
    | expresion '/' expresion
    | expresion '%' expresion

```

```

    | expresion OR expresion
    | expresion AND expresion
    | expresion GREATER_EQUAL expresion
    | expresion LESS_EQUAL expresion
    | expresion EQUAL expresion
    | expresion NOT_EQUAL expresion
    | expresion '<' expresion
    | expresion '>' expresion
    ;
r_expresion: /* Lambda */
    | ';' expresion r_expresion
    ;
incremento: IDENTIF '=' expresion '+' expresion
    | IDENTIF '=' expresion '-' expresion
    ;
termino:    operando
    | '+' operando %prec UNARY_SIGN
    | '-' operando %prec UNARY_SIGN
    ;
operando:   IDENTIF
    | NUMBER
    | IDENTIF '[' expresion ']'
    | IDENTIF '(' ')'
    | IDENTIF '(' expresion r_expresion ')'
    | '(' expresion ')'
    ;

```

3. Conclusiones