Manual Técnico

Traductor PY

Josue Guillermo Orellana Cifuentes
201801366 Facultad de Ingeniería USAC

La gramática que se muestra a continuación es la que se utilizó para el análisis sintáctico:

```
DECLARACION_ASIGNACION ->     TIPO_DATO LISTA_ID
                              DECLARAICION_ASIGNACION_P

DECLARAICION_ASIGNACION_P ->  ;
                              |= EXPRESION;

ASIGNACION ->      id = EXPRESION;

LISTA_ID ->        id LISTA_ID_P

LISTA_ID_P ->      , id LISTA_ID_P
                   | ε

EXPRESION ->       TERMINO EXPRESION_P

EXPRESION_P ->     + TERMINO EXPRESION_P
                   |- TERMINO EXPRESION_P
                   |ε

TERMINO ->         FACTOR TERMINAL_P

TERMINO_P ->       * FACTOR TERMINO_P
                   |/ FACTOR TERMINO_P
                   |ε

FACTOR ->          (EXPRESION)
                   |número
                   |id LLAMADA_FUNCION
                   |cadena
                   |true
                   |false

LLAMADA_FUNCION ->      (ARGUMENTOS)
                        | ε

LLAMADA_METODO -> id (ARGUMENTOS);

ARGUMENTOS ->      EXPRESION ARGUMENTOS_P

ARGUMENTOS_P ->    , ARGUMENTOS
                   | ε
```

```
PARAMETROS ->     TIPO_DATO id PARAMETROS_P
PARAMETROS_P ->   , PARAMETROS
                  | ε
DECLARACION_METODO -> void id (PARAMETROS) {SENTENCIAS }
RETURN_METODO ->  return;
                  | ε
DECLARACION_FUNCION ->  TIPO_FUNCION id (PARAMETROS) {SENTENCIAS }
DECLARACION_MAIN ->     void main () {SENTENCIAS}
DECLARACION_IF ->       if (CONDICIONAL_IF) {SENTENCIAS} DECLARACION_IF_P
CONDICIONAL_IF ->       OPERADOR_NOT EXPRESION_RELACIONAL OPERADOR_LOGICO
DECLARACION_IF_P ->     else DECLARACION_ELSE
                        | ε
OPERADOR_NOT ->         !
                        | ε
DECLARACION_ELSE ->     {SENTENCIAS}
                        | DECLARACION_IF
OPERADOR_RELACIONAL ->        SIGNO_OPERADOR_RELACIONAL EXPRESION
                              | ε
SIGNO_OPERADOR_RELACIONAL ->  >
                              | <
                              | >=
                              | <=
                              | ==
                              | !=
OPERADOR_LOGICO ->      SIGNO_OPERADOR_LOGICO   CONDICIONAL_IF
                        | ε
SIGNO_OPERADOR_LOGICO ->      &&
                              | ||
DECLARACION_SWITCH ->   switch (id){DECLARACION_CASE}
```

```
DECLARACION_CASE ->     case EXPRESION: SENTENCIAS break;
                        DECLARACION_CASE_P

DECLARACION_CASE_P ->   DECLARACION_CASE

                        | default: SENTENCIAS break;

DECLARACION_FOR ->      for (SENTENCIA EXPRESION
                        SIGNO_OPERADOR_RELACIONAL EXPRESION; id
                        OPERADOR_INC_DEC) {SENTENICAS}

OPERADOR_INC_DEC ->      ++;

                        | --;

DECLARACION_WHILE ->    while (CONDICIONAL_IF) {SENTENCIAS}

DECLARACION_DO_WHILE ->       do {SENTENCIAS} while (CONDICIONAL_IF);

IMPRIMIR ->       Console . Write (IMPRIMIR_P);

IMPRIMIR_P ->     ‘cadena ‘

                  | ”cadena ”

COMENTARIO ->     // cadena \n

                  |/* cadena */

EXPRESION_RELACIONAL ->       TERMINO_RELACIONAL EXPRESION_RELACIONA_P

EXPRESION_RELACIONAL_P ->     + TERMINO_RELACIONAL EXPRESION_RELACION_P

                              |- TERMINO_RELACIONAL EXPRESION_RELACION_P

                              |* TERMINO_RELACIONAL EXPRESION_RELACION_P

                              |/ TERMINO_RELACIONAL EXPRESION_RELACION_P

                              | ε


TERMINO_RELACIONAL -> FACTOR_RELACIONAL TERMINO_RELACIONAL_P

TERMION_RELACIONAL_P ->       > FACTOR_RELACIONAL TERMION_RELACIONAL_P
                              | < FACTOR_RELACIONAL TERMION_RELACIONAL_P

                              | >= FACTOR_RELACIONAL TERMION_RELACIONAL_P

                              | <= FACTOR_RELACIONAL TERMION_RELACIONAL_P

                              | == FACTOR_RELACIONAL TERMION_RELACIONAL_P

                              | != FACTOR_RELACIONAL TERMION_RELACIONAL_P

                              | ε
```

FACTOR_RELACIONAL -> (EXPRESION_RELACIONAL)

|número

|id LLAMADA_FUNCION

|cadena

|true

|false

PRODUCCION_BREAK -> break;

| ε

PRODUCCION_CONTINUE -> continue;

| ε

LAS PRODUCCIONES DE BREAK Y CONTINUE SOLO PUEDEN SER PRODUCIDAS SI SE ENCUENTRAN DENTRO DE CICLOS DE REPETICIÓN.

PRODUCCION_RETURN -> return PRODUCCION_RETURN_P

| ε

PRODUCCIÓN_RETURN_P -> EXPRESION_RELACIONAL ;

| ε ;

LA PRODUCCION DE RETURN SOLO SE PUEDE PRODUCIR DENTRO DE MÉTODOS Y FUNCIONES.