Gramática ASCENDENTE

```
def p_init(t):
    'init  :    MAIN DOSPUNTOS listainstrucciones'

def p_lista_instrucciones(t):
    'listainstrucciones :   listainstrucciones simpleinstrucciones'

def p_lista_instrucciones_simple(t):
    'listainstrucciones : simpleinstrucciones'
def p_simpleinstrucciones(t):
    '''simpleinstrucciones    : declaracion
                              | asignacion
                              | ifcomando
                              | exit
                              | print
                              | etiqueta
                              | goto
                              | unset'''
def p_declaracion(t):
    'declaracion   :    variable PTCOMA'

def p_asignacion(t):
    'asignacion :   variable IGUAL tipo PTCOMA'


def p_tipo(t):
    '''tipo :   instruccionregistro
            |   conversion
            |   arreglo'''

def p_variable(t):
    '''variable :   TEMPORALES
                |   PARAMETROS
                |   DEVOLUCIONES
                |   RETORNONIVEL
                |   PILA
                |   SP'''
```

```
def p_variable_arreglo(t):
    '''variable :   TEMPORALES listadimension
                |   PARAMETROS listadimension
                |   DEVOLUCIONES listadimension
                |   RETORNONIVEL listadimension
                |   PILA listadimension
                |   SP listadimension'''

def p_instruccionregistro(t):
    '''instruccionregistro  : registro MAS registro
                            | registro MENOS registro
                            | registro POR registro
                            | registro DIVIDIDO registro
                            | registro RESIDUO registro
                            | registro ANDLOGICA registro
                            | registro ORLOGICA registro
                            | registro XOR registro
                            | registro ANDBIT registro
                            | registro ORBIT registro
                            | registro XORBIT registro
                            | registro SHIFTIZQ registro
                            | registro SHIFTDER registro
                            | registro IGUALIGUAL registro
                            | registro NOIGUAL registro
                            | registro MAYORIGUAL registro
                            | registro MENORIGUAL registro
                            | registro MAYOR registro
                            | registro MENOR registro'''
def p_instruccion_registrounico(t):
    '''instruccionregistro  : registro'''

def p_registro_parentesis(t):
    '''registro :   PARIZQ registro PARDER'''

def p_instruccionregistro_diferentes(t):
    '''registro  : ABS PARIZQ registro PARDER
                            | NOTLOGICA registro
                            | NOTBIT registro
                            | MENOS registro %prec UMENOS
                            | ANDBIT registro'''
def p_nuevo_(t):
    'registro   :   readisntr'
```

```
def p_registro(t):
    '''registro :   ENTERO
                |   DECIMAL'''

def p_registro_cadena(t):
    '''registro :   CADENA'''

def p_registro_acceso(t):
    '''registro :   acceso'''

def p_acceso(t):
    '''acceso   :   variable'''


def p_lista_dimension(t):
    '''listadimension    :   listadimension dimension'''

def p_lista_dimnension2(t):
    '''listadimension   :   dimension'''

def p_dimension(t):
    '''dimension    :   CORCHETEIZQ registro CORCHETEDER'''

def p_conversion(t):
    'conversion   : PARIZQ eltipo PARDER registro'

def p_arreglo(t):
    'arreglo    :   ARRAY PARIZQ PARDER'

def p_eltipo(t):
    '''eltipo   :   FLOAT
                |   INT
                |   CHAR'''
def p_exit(t):
    'exit   :   EXIT PTCOMA'

def p_print(t):
    'print  : PRINT PARIZQ registro PARDER PTCOMA'

def p_etiqueta(t):
    'etiqueta   :  ETIQUETA DOSPUNTOS' #
def p_goto(t):
    'goto   :   GOTO ETIQUETA PTCOMA' #
```

```
def p_read(t):
    'readisntr   :   READ PARIZQ PARDER '

def p_unset(t):
    'unset  :   UNSET PARIZQ acceso PARDER PTCOMA'

def p_if(t):
    'ifcomando :    IF PARIZQ instruccionregistro PARDER GOTO ETIQUETA PTCOM
A'
```

EXPLICACION

Para este tipo de gramática ascendente se hizo uso de la recursividad y del trabajo con ambigüedad ya que este tipo de analizador lo permite, el flujo de la gramática es muy simple,

Se tiene una lista de instrucciones que contienen todas las acciones que el lenguaje AUGUS nos permite utilizar, dentro de cada una se desglosa la estructura de cada uno

'simpleinstrucciones' es la encargada de llamar a cada instrucción y así generar las listas, las instrucciones o llamados de funciones propias del lenguaje se desglosan por separado y sin tener una estructura muy grande

La parte mas importante de la gramática es instrucciones registro

```
  '''instruccionregistro  : registro MAS registro
                          | registro MENOS registro
                          | registro POR registro
```

Esta producción deriva en cada una de las operaciones aritméticas, lógicas y más del lenguaje, esta podría ser recursiva para así generar operaciones complejas, pero sin embargo el lenguaje no lo permite por lo que solo se manejan operaciones entre dos operados.

La segunda parte más importante la encontramos en la variable y en los tipos que puede tomar y el modo de acceder a sus dimensiones

```
def p_variable_arreglo(t):
    '''variable :   TEMPORALES listadimension
                |   PARAMETROS listadimension
                |   DEVOLUCIONES listadimensic
                |   RETORNONIVEL listadimensic
                |   PILA listadimension
                |   SP listadimension'''
```

Gramática DESCENDENTE

```
def p_init(t):
    'init  :    MAIN DOSPUNTOS listainstrucciones'

def p_lista_instrucciones(t):
    'listainstrucciones :   simpleinstrucciones listainstruccionesp'

def p_lista_instrucciones_prima(t):
    'listainstruccionesp : simpleinstrucciones listainstruccionesp'

def p_lista_instrucciones_epsilon(t):
    'listainstruccionesp    :   '


def p_simpleinstrucciones(t):
    '''simpleinstrucciones    : declaracion
                              | asignacion
                              | ifcomando
                              | exit
                              | print
                              | etiqueta
                              | goto
                              | unset'''

def p_declaracion(t):
    'declaracion   :    variable PTCOMA'

def p_asignacion(t):
    'asignacion :   variable IGUAL tipo PTCOMA'


def p_tipo(t):
    '''tipo :   instruccionregistro
            |   conversion
            |   arreglo'''

def p_variable(t):
    '''variable :   TEMPORALES
                |   PARAMETROS
```

```
                |   DEVOLUCIONES
                |   RETORNONIVEL
                |   PILA
                |   SP'''

def p_variable_arreglo(t):
    '''variable :   TEMPORALES listadimension
                |   PARAMETROS listadimension
                |   DEVOLUCIONES listadimension
                |   RETORNONIVEL listadimension
                |   PILA listadimension
                |   SP listadimension'''

def p_instruccionregistro(t):
    '''instruccionregistro  : registro operacion registro'''

def p_operacion(t):
    '''operacion            :  MAS
                            |  MENOS
                            |  POR
                            |  DIVIDIDO
                            |  RESIDUO
                            |  ANDLOGICA
                            |  ORLOGICA
                            |  XOR
                            |  ANDBIT
                            |  ORBIT
                            |  XORBIT
                            |  SHIFTIZQ
                            |  SHIFTDER
                            |  IGUALIGUAL
                            |  NOIGUAL
                            |  MAYORIGUAL
                            |  MENORIGUAL
                            |  MAYOR
                            |  MENOR '''


def p_instruccion_registrounico(t):
    '''instruccionregistro  : registro'''

def p_registro_parentesis(t):
    '''registro :   PARIZQ registro PARDER'''
```

```
def p_instruccionregistro_diferentes(t):
    '''registro  : ABS PARIZQ registro PARDER
                            | NOTLOGICA registro
                            | NOTBIT registro
                            | MENOS registro %prec UMENOS
                            | ANDBIT registro'''
def p_nuevo_(t):
    'registro   :   readisntr'

def p_registro(t):
    '''registro :   ENTERO
                |   DECIMAL'''

def p_registro_cadena(t):
    '''registro :   CADENA'''

def p_registro_acceso(t):
    '''registro :   acceso'''

def p_acceso(t):
    '''acceso   :   variable'''


def p_lista_dimension(t):
    '''listadimension    :   dimension listadimensionp'''

def p_lista_dimension_prima(t):
    ''' listadimensionp :   dimension listadimensionp'''

def p_listadimension_epsilon(t):
    '''listadimensionp   :  '''

def p_dimension(t):
    '''dimension    :   CORCHETEIZQ registro CORCHETEDER'''
def p_conversion(t):
    'conversion   : PARIZQ eltipo PARDER registro'



def p_eltipo(t):
    '''eltipo   :   FLOAT
                |   INT
                |   CHAR'''

def p_arreglo(t):
```

```
    'arreglo    :   ARRAY PARIZQ PARDER'

def p_exit(t):
    'exit   :   EXIT PTCOMA'
    'print  : PRINT PARIZQ registro PARDER PTCOMA'

def p_etiqueta(t):
    'etiqueta   :  ETIQUETA DOSPUNTOS'

def p_goto(t):
    'goto   :   GOTO ETIQUETA PTCOMA'


def p_read(t):
    'readisntr   :   READ PARIZQ PARDER '

def p_unset(t):
    'unset  :   UNSET PARIZQ acceso PARDER PTCOMA'

def p_if(t):
    'ifcomando :    IF PARIZQ instruccionregistro PARDER GOTO ETIQUETA PTCOM
A'
```

EXPLICACION:

Las diferencias con la gramática DESC son muy reducidas, la única diferencia es la recursividad que en este caso se elimino y la factorización

```
def p_lista_instrucciones(t):
    'listainstrucciones :   simpleinstrucciones listainstruccionesp'

def p_lista_instrucciones_prima(t):
    'listainstruccionesp : simpleinstrucciones listainstruccionesp'

def p_lista_instrucciones_epsilon(t):
    'listainstruccionesp  :   '
```

Como podemos ver se tiene la misma estructura, pero se aplicaron las reglas de recursividad por la izquierda y también se cambio el acceso a las instrucciones del lenguaje

```
def p_instruccionregistro(t):
    '''instruccionregistro  : registro operacion registro'''

def p_operacion(t):
    '''operacion            :  MAS
                            |  MENOS
                            |  POR
                            |  DIVIDIDO
                            |  RESIDUO
                            |  ANDLOGICA
                            |  ORLOGICA
                            |  XOR
                            |  ANDBIT
                            |  ORBIT
                            |  XORBIT
                            |  SHIFTIZQ
                            |  SHIFTDER
                            |  IGUALIGUAL
                            |  NOIGUAL
                            |  MAYORIGUAL
                            |  MENORIGUAL
                            |  MAYOR
                            |  MENOR '''
```

Siendo la no terminal operación el que contiene los diferentes tipos de operaciones entre dos operandos que podemos utilizar