Sintaxis del lenguaje PL/0 en BNF

```
<programa> ::= <bloque> "."
<bloque> ::= <declaracion-const> <declaracion-var> <declaraciones-procedures> <proposicion>
<declaracion-const> ::= "CONST" <identificador> "=" <numero> <declarar-mas-idents-igual-numero> ";" | ϵ
<declarar-mas-idents-igual-numero> ::= "," <identificador> "=" <numero> <declarar-mas-idents-igual-numero> | ϵ
<declaracion-var> ::= "VAR" <identificador> <declarar-mas-idents> ";" | ϵ
<declarar-mas-idents> ::= "," <identificador> <declarar-mas-idents> | ϵ
<declaraciones-procedures> ::= "PROCEDURE" <identificador> ";" <bloque> ";" <declaraciones-procedures> | ϵ
<proposicion> ::= <identificador> ":=" <expresion> |
                  "CALL" <identificador> |
                  "BEGIN" <proposicion> <mas-propos> "END" |
                  "IF" <condicion> "THEN" <proposicion> |
                  "WHILE" <condicion> "DO" <proposicion> | ϵ                             Obs.:  ϵ = cadena vacía
<mas-propos> ::= ";" <proposicion> <mas-propos> | ϵ
<condicion> ::= "ODD" <expresion> | <expresion> <operador-relacional> <expresion>
<operador-relacional> ::= "=" | "<>" | "<" | "<=" | ">" | ">="
<expresion> ::= <signo-unario> <termino> <mas-terminos>
<signo-unario> ::= "+" | "-" | ϵ
<termino> ::= <factor> <mas-factores>
<mas-terminos> ::= "+" <termino> <mas-terminos> | "-" <termino> <mas-terminos> | ϵ
<factor> ::= <identificador> | <numero> | "(" <expresion> ")"
<mas-factores> ::= "*" <factor> <mas-factores> | "/" <factor> <mas-factores> | ϵ           Reglas para el Scanner:
```

```
<identificador> ::= <letra> <mas-caracteres>
<numero> ::= <digito> | <digito-mayor-que-cero> <mas-digitos>
<letra> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" |
            "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
<mas-caracteres> ::= <caracter-identificador> <mas-caracteres> | ϵ
<caracter-identificador> ::= <letra> | <digito>
<digito> ::= "0" | <digito-mayor-que-cero>
<digito-mayor-que-cero> ::= "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
<mas-digitos> := <digito> <mas-digitos> | ϵ
```