|  |  |  |
| --- | --- | --- |
| 1 | Dígito |  |
| 2 | Entero |  |
| 3 | Signo |  |
| 4 | Entero con Signo |  |
| 5 | Mayúscula |  |
| 6 | Minúscula |  |
| 7 | Letra |  |
| 8 | Letras |  |
| 9 | Identificador |  |
| 10 | Variable |  |
| 11 | Termino Aritmético | Invocacion | Identificador | (ExprecionAritmetica) |
| 12 | Expresión Aritmética | <TerminoAritmetico> < OperadorAritmetico> < ExpresionAritmetica> | TerminoAritmetico |
|  | Identificador | <Identificador>::= <IdVariable>|<IdClase>|<IdMetodo> |
| 13 | Sentencia | <Sentencia>::= <SentenciaAsignacion>|<SentenciaIf>|<SentenciaWhile>|<SentenciaFor>|<LlamadaMetodo> |
| 14 | Lista de Sentencias | <ListaSentencias> ::= <Sentencia><ListaSentencias> | <Sentencia> |
| 15 | Sentencia Si | <SentenciaIf>::= ”if(“ <Condición> ”)” “{“ <ListaSentencias> “}” | “if(“ <Condición> ”)” “{“ <ListaSentencias> “}” “else{“ <ListaSentencias> “}” |
| 16 | Sentencia Mientras | <SentenciaWhile>::= ” while (“<Condicion>”)” ”{“ <ListaDeSentencias>”}” |
| 17 | Sentencia Hacer  Mientras | <SentenciaDo>::= ” do {“ <ListaDeSentencias>”}” ”(“<Condición>”)” |
| 18 | Sentencia Para | <SentenciaFor>::= ” for (“<Asignación> ”;” <Condición>”;”<Asignación> “) {“ <ListaDeSentencias>”}” |
| 19 | Factor | <Factor> ::= <Expresión> | <Identificador> |
| 20 | Termino | <Termino>::= <Factor> | <Termino> “\*” <Factor>| <Termino> “/” <Factor> |
| 21 | Expresión | <Expresión> ::= <Término> |<Expresión> “+” <Termino>|<Expresión> “-” <Termino>|<Expresión> <OperadorRelacional><Termino>>|<Expresión> <OperadorLogico>”<Termino> |
| 22 | Sentencia de Asignación | <SentenciaAsignacion>::= <IdVariable> “=”<Expresion> |
| 23 | Condición Simple | <CondicionSimple>::= “!”<CondicionSimple> | <Expresion> <OperadorRelacional><Expresion> | “(“ <Condicion> “)” | verdadero | falso |
| 24 | Condición | <Condición> ::= < CondicionSimple > | <Condición> <OperadorRelacional>< Condición > |
| 25 | Lista Parámetros | <ListaParametros>::= <Identificador>|<ListaParametros> “,” <Identificador> |
| 26 | DefiniciónMetodo | <DefiniciónMetodo>::= <IdMetodo> “(“ [<ListaParametros>] “){” <ListaSentencias> “}” |
| 27 | Lista Expresiones | <ListaExpresiones> ::= <Expresion>”,” <ListaExpresiones> | <Expresion> |
| 28 | Llamada Función | <LlamadaFuncion>::= <Identificador> “(“ [<ListaExpresiones>] “)” |
| 29 | Lista Definiciones  Metodo | <ListaDefinicionesMetodo>::= <ListaDefinicionesMetodo> <DefinicionMetodo> | <DefinicionMetodo> |
| 30 | Programa | <Programa>::= <ListaSentencias> | [<ListaDefinicionesMetodo>] |

**Operadores Aditivos**

<Aditivo>::=<Aditivo>+<Aditivo>|<Numero>

**Operadores Multiplicativos**

<Multiplicativo>::=<Multiplicativo>\*<Multiplicativo>|<Numero>

**Operadores Relacionales**

<OperadorRelacional>::="<"|">"|"<="|">="|"=="|"!="

<Relacional>::=<Relacional><OperadorR><Relacional>|<Numero>

<OperadorLogico>::="&&"|"||"|"!"

<Logico>::=<Logico><OperadorLogico><Logico>|<Numero>

**Operador Asignación**

<Combinacion>::=<Numero>|<Variable>

<OperadorAsignacion>::="="|"-="|"+="

<Asignacion>::=<Variable><OperadorAsignacion><Combinacion>

**FUNCION**

*<Funcion> ::= <TipoRetorno> identificadorMetodo "("[<ListaParametros>]")" <LlaveApertura> <BloqueSentencias> <LlaveCierre>*

* **INCREMENTO DECREMENTO**

**<IncrementoDecremento> ::=** *identificadorVariable ++ | identificador de variable --*

**ARREGLO**

*<Arreglo> ::= list identificadorVariable ":" <TipoDato> "=" "[" <ListaArgumentos>"]" ";"*

**UNIDAD DE COMPILACION**

*<UnidadDeCompilacion> ::= <ListaFunciones>*

**SENTENCIA**

*<sentencia> ::= <Decisión> | <Ciclo> | <Impresion> | <Lectura> | <Retorno> |<InvocasionFuncion> |<IncrementoDecremento> | <DeclaracionVariable>*

* **ES TIPO DATO**

*<esTipoDato> ::= int|float|chat|boolean|string*

* **TIPO RETORNO**

*<tipoRetorno> ::= int| decimal| string|char|bool|void*

**RETORNO**

*<Retorno> ::= return":" <Expresion>*

* **LISTA PARAMETRO**

*<ListaParametros> ::=<Parametro>[","<ListaParametros>]*

* **MUTABLE**

**<Mutable> ::= const | let**

**DECLARACION VARIABLE (mutable o inmutable)**

*<DeclaracionVariable>::= <mutable> <IdentificadorVariable> : <TipoDato>["=" <Expresion>]*

* **PARAMETRO**

*<Parametro> ::= identificadorVariable":"<TipoDato>*

**BLOQUE SENTENCIA**

*<BloqueSentencias> ::= "{" [<ListaSentencia>] "}"*

**ES LISTA ARGMENTOS**

*\* <ListaArgumentos> ::= <Argumento>["," <ListaArgumentos>]  
\* <Argumento> ::= <Expresion>*

**LISTA SENTENCIA**

*<ListaSentencias> ::= <Sentencia>[<ListaSentencias>]*

**CICLO**

*<Ciclo> ::= while “(“<ExpresionLogica>”)” <Bloque>*

DECISIÓN

*<Decision> ::= if <ExpresionLogica> <Bloque> [else <Bloque>]*

ASIGNACION

*<Asignacion> ::= <IdVariable> <OperadorAsignacion> <Expresion>*

EXPRESION

*<Expresion> ::= <ExpresionAritmetica> | <ExpresionRelacional> | <ExpresionLogica>*

EXPRESION ARITMETICA

*<ExpresionAritmetica> ::= "("<ExpAritmetica>")" [operadorAritmetica <ExpAritmetica>]|<ValorNumerico>| <InvocacionFuncion>  
\* | <Identificador>*

IMPRESIÓN (datos o variales)

*<Impresion> ::= print "(" <Expresion> “)”*

CICLO (mientas o for)

*\* <Ciclo> ::= while <ExpresionLogica> “{“<Bloque>”]”*

DECISION (if - else)

*\* <Decision> ::= if <ExpresionLogica> <Bloque> [else <Bloque>]*

INVOCASION FUNCION

*\* <InvocacionFuncion> ::= identificadorFuncion "("<ListaArgumentos>")" ";"*

LAECTURA (de datos)

*<Lectura> ::= input"(" [cadena] ")"*

EXPRESION RELACIONAL

*\* <ExpresionRelacional> ::= <ExpresionAritmetica> operacionRelacional <ExpresionAritmetica> | false | true*

EXPRESION LOGICA

*\* ExpresionLogica> ::= "!" <ExpresionRelacional> | <ExpresionRelacional> || <ExpresionRelacional> |  
\* <ExpresionRelacional> && <ExpresionRelacional> |  
\* <ExpresionRelacional>*

EXPRESION CADENA

*\*<ExpresionCadena> ::= cadena["+" <Expresion>] | identificador]*

VALOR NUMERICO

*\* <ValorNumerico> ::= [<signo>] decimal | [<signo>] entero | [<signo>] identificador*

OPERADOR ARITMETICO

<OperadorAritmetico> ::= + | - | / | \* | %

**Operadores Relacionales**

<OperadorRelacional> ::= "<" | ">" | "<=" | ">=" | "==" | "!="