**BNF’S Lenguaje Trueno**

Jeffrey Alexander Vargas, Kevin Valencia Romero

<UnidadDeCompilacion> ::= <ListaFunciones>

<Invocacion> ::= identificador "(" [<ListaArgumentos>] ")" ";"

<ListaArgumentos> ::= <Expresion> ["," <ListaArgumentos> ]

<ListaFunciones> ::= <Funcion>[<ListaFunciones>]

<Funcion> ::= function identificador "$"[<ListaParametros>]"$" [":"<TipoRetorno>] <BloqueSentencias>

<TipoRetorno> ::= \_entero | \_decimal | \_real | \_cadena | \_caracter | void

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

<Parámetro> ::= identificador":"<TipoDato>

<TipoDato> ::= \_entero | \_decimal | \_real | \_cadena | \_caracter

<TipoVariable> ::= mutable | inmutable

<ListaVariables> ::= <Variable> [“,” <ListaVariables>]

<Variable> ::= Identificador [ operadorAsignacion <Expresion>]

<ListaIdentificadores> ::= Identificador [“,”<ListaIdentificadores> ]

<BloqueSentencias> ::= "$" [<ListaSentencias>] "$"

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

<Sentencia> ::= <Decision> | <Ciclo> | <Impresion> | <Lectura> | <Asignacion> | <DeclaracionVariable> | <Retorno> | <InvocacionFuncion> | <Arreglo>

<Decision> ::= siEs "$" <ExpresionLógica> "$" <BloqueSentencias>

<Ciclo> ::= &iterar& <ExpresionLogica> <Bloque>

<Impresión> ::= toPrint “$” Identificador “$”

<Lectura> ::= toRead “$” Identificador “$”

<Asignacion> ::= identificador operadorAsignacion <Expresion> ";" | identificador operadorAsignacion <InvocarFuncion> ";"

<DeclaracionVariable> ::= var <ListaVariables>":"<TipoDato>";"

<Retorno> ::= salida <Expresion> | null

<Arreglo> ::= arrangement ”[“ TipoDato “]” identificador [“<” <ListaDatos> “>”]

<ListaDatos> ::= <TipoDato> [“<ListaDatos>”]

<Expresiones> ::= <Expresion> [<Expresiones>]

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

< ExpresionArimetica > ::= <ExpresionArimetica> operadorAritmetico <ExpresionArimetica> | "$"< ExpresionArimetica >"$" | <ValorNumerico>

<ValorNumerico> ::= [<Signo>] real | [<Signo>] entero | [<Signo>] identificador

<ExpresionRelacional> ::= <ExpresionRelacional> operadorRelacional <ExpresionRelacional>

<ExpresionLogica> ::= <ExpresionLogica> OperadorLogico <ExpresionLogica> | <ExpresionRelacional>