Parser -> Program\_Code $

Program\_Code -> Lpp\_Program Program\_Code  
 | Html Program\_Code  
 |ϵ

Lpp\_Program -> Program\_Header   
 Statement\_List

Program\_Header -> Types\_List  
 |Declare\_Variables   
 |Methods\_List  
 | ϵ

Types\_List -> Types\_Structure Types\_List   
 |ϵ

Types\_Structure -> TIPO ID ES Type   
 |REGISTRO ID Declare\_Variables FIN REGISTRO

Type -> entero  
 |real  
 |booleano  
 |caracter  
 |cadena [ const\_entero ]  
 | ARREGLO [ Array\_Size ] DE Type  
 |ARCHIVO DE Type  
 |ID

Declare\_Variables -> Variables\_Group Declare\_Variables  
 | ϵ

Variables\_Group -> Type ID ID\_List

ID\_List-> , ID ID\_List  
 | ϵ

Array\_Size -> const\_entero Integer\_List

Integer\_List -> , const\_entero Integer\_List  
 | ϵ

Methods\_List -> Method   
 Declare\_Variables   
 Program\_Body  
 Methods\_List  
 | ϵ

Method -> funcion ID Params\_List : Type  
 | procedimiento ID Params\_List

Params\_List -> ( Declare\_Params )  
 | ϵ

Declare\_Params -> Param Param\_Group

Param\_Group -> , Param Param\_Group  
 | ϵ

Param -> var Type ID  
 |Type ID

Statement\_List -> Statement Statement\_List  
 | ϵ

Statement -> Statement\_Si  
 |Statement\_Mientras   
 |Statement\_Llamar   
 |Statement\_Case  
 |Statement\_Abrir\_Archivo   
 |Statement\_Escribir\_Archivo   
 |Statement\_Leer\_Archivo   
 |Statement\_Para   
 |Statement\_Repita   
 |Statement\_ Assignment  
 |Statement\_Escriba  
 |Statement\_Cerrar\_Archivo

Statement\_Si -> si Condition entonces  
 Statement\_List  
 fin si  
 | si Condition entonces  
 Statement\_List  
 sino  
 Statement\_List  
 fin si

Statement\_Para -> para Variable <- Expression hasta Expression haga  
 Statement\_List  
 fin para

Variable -> Simple\_Variable Compuest\_Variable

Compuest\_Variable -> . Variable  
 | ϵ

Simple\_Variable -> ID String\_Variable

String\_Variable -> [ Expression\_List ]  
 | ϵ

Expression\_List -> Expression Expression\_Group  
 | ϵ

Expression\_ Group -> , Expression Expression\_Group  
 | ϵ

Statement\_Mientras -> mientras Expression haga  
 Statement\_List  
 fin mientras

Statement\_Repita -> repita   
 Statement\_List  
 hasta Expression

Statement\_Llamar -> llamar ID Params\_List

Statement\_ Assignment -> Variable <- Expression

Statement\_Case -> caso Variable  
 Case\_List  
 Sino\_Case  
 fin caso

Case\_List -> Define\_Case Case\_Group

Define\_Case -> Literal\_List : Statements\_List

Case\_Group -> Define\_Case Case\_Group   
 | ϵ

Literal\_List -> Literal Literal\_Group

Literal\_Group -> , Literal Literal\_Group  
 | ϵ

Literal-> const\_entero  
 |const\_real  
 |const\_caracter  
 |const\_cadena

Sino\_Case-> sino : Statement\_List  
 | ϵ

Statement\_Escriba -> escribe Expression\_List

Statement\_Abrir\_Archivo -> abrir Expression como Variable Operation\_List

Operation\_List -> para Operation , Operation  
 | ϵ

Operation-> lectura   
 |escritura

Statement\_Cerrar\_Archivo -> cerrar Variable

Statement\_Escribir\_Archivo -> escribir Expression , Expression\_List

Statement\_Leer\_Archivo -> leer Variable , Variable Variable\_List

Variable\_List -> , Variable Variable\_List   
 | ϵ

Expression -> factor Expression\_With\_Operators

Expression\_With\_Operators -> Operator Expression\_With\_Operators   
 | ϵ

Operator -> +  
 | -  
 | \*  
 | /  
 | ^  
 | mod  
 | div  
 | y  
 | o

Factor -> Variable  
 | ID  
 |const\_entero  
 |const\_cadena  
 |const\_caracter  
 |const\_real  
 |const\_entero  
 |const\_entero  
 |const\_entero