GRAMÁTICA LL(1) EstructHasta EstructIncremento G(N, T, S, P): desde 0 hasta **⋄** N: **❖** T: o Codigo **PRprincipal** SimboloInicial 0 (Texto) 0 DeclaVariable 0 Incrementable 0 0 VariablesL **PRDatoPrimi** 0 id TokId 0 idVar ++ 0 valor 0 **TipoValor** 0 Suma 0 0 Suma' 0 cadena entero 0 Multi decimal 0 Multi' 0 booleano 0 Pot 0 char Pot' 0 NumE 0 Unar 0 **PRImprimir** 0 Elem 0 Impresión 0 ValImprimir **Imprimir** 0 Append 0 0 **PRLeer** Lectura 0 **PRSI** 0 ValLeer 0 Condicion PRSINO_SI DescCondicion **PRSINO** 0 CuerpoDescCond 0 Numero 0 Cond2 0 OpRel ValCondicion PRBooleana 0 DescValCondicion ! 0 Condicionante OpLogico o SigDif **PRHACER** CondicionantePlus **PRMIENTRAS** 0 0 **PRDESDE** CicloM 0 **PRHASTA** 0 CicloHM **PRINCREMENTO** DescCicloHM NumE 0 CicloF **❖** S: Codigo EstructCicloF

EstructDesde

❖ P: Producción

Codigo	\rightarrow	PRprincipal SimboloInicial Texto
SimboloInicial	\rightarrow	() {
Texto	→	VariablesL Texto DeclaVariable Texto Impresión Texto Lectura Texto Condicion Texto CicloM Texto CicloHM Texto CicloF Texto 1
DeclaVariable	→	TokId = TipoValor; TokId Incrementable;
Incrementable	\rightarrow	++
VariablesL	>	PRDatoPrimi idVar id
		, idVar id
id	\rightarrow	, id var id ;
idVar	\rightarrow	TokId valor
iu v ai		TOKIU VAIOI
valor	>	= TipoValor e
TipoValor	→	cadena Suma decimal booleano char
Suma	\rightarrow	Multi Suma'

Suma' Multi Suma' - Multi'		ı	
Multi Pot Multi' Pot Multi' Pot Multi' Pot Multi' Pot Pot Pot Pot' Pot	Suma'	\rightarrow	- Multi Suma'
Multi' Pot Multi' Pot Multi' Pot Multi' Pot Multi' Pot	N /14:		
Multi' → /Pot Multi' e Pot → Unar Pot' Pot' → ^Pot e Unar → -Elem Elem Elem → Num (Suma) Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Imprimir → + Imprimir Append 1); Lectura → PRLeer ValLeer; ValLeer → (Tokid)	Multi	フ	
le		_	
Pot → Unar Pot' Pot' → Pot e Unar → IElem Elem Flem Elem → TokId Num (Suma) Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Imprimir → Imprimir Append Append → Imprimir Append 1); Lectura → PRLeer ValLeer; ValLeer → (TokId)	Multi'	\rightarrow	/ Pot Multi'
Pot' Pot Pot			
Pot	Pot	\rightarrow	Unar Pot'
Pot Proceedings Pot Proceedings Pot Proceedings Pot	D 42		^ Pot
Unar → Elem Elem Elem → Tokld Num (Suma) Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Cadena Num Char	Pot	7	l e
Elem			
Elem → Tokld Num (Suma) Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Imprimir → Tokld cadena Num char Append → + Imprimir Append); Lectura → PRLeer ValLeer; ValLeer → (Tokld)	Unar	\rightarrow	
Elem → Num (Suma) Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Imprimir → TokId cadena Num char Append → + Imprimir Append); Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Imprimir → TokId cadena Num char Append → + Imprimir Append): Lectura → PRLeer ValLeer; ValLeer → (TokId)	Flem	→	
Impresión → PRImprimir ValImprimir ValImprimir → (Imprimir Append Imprimir → TokId cadena Num char Append → Imprimir Append	Liciii		
ValImprimir → (Imprimir Append Imprimir → TokId cadena Num char Append → Imprimir Append			(Suilla)
ValImprimir → (Imprimir Append Imprimir → TokId cadena Num char Append → Imprimir Append			
ValImprimir → (Imprimir Append Imprimir → TokId cadena Num char Append → Imprimir Append	T '/		
Imprimir TokId cadena Num char Append Himprimir Append ; ; Lectura PRLeer ValLeer ; ValLeer (TokId)	Impresion	7	PRImprimir Valimprimir
Imprimir TokId cadena Num char Append Himprimir Append ; ; Lectura PRLeer ValLeer ; ValLeer (TokId)			
Imprimir TokId cadena Num char Append Himprimir Append ; ; Lectura PRLeer ValLeer ; ValLeer (TokId)			
Imprimir → cadena Num char Append → + Imprimir Append); Lectura → PRLeer ValLeer; ValLeer → (TokId)	ValImprimir	\rightarrow	(Imprimir Append
Imprimir → cadena Num char Append → + Imprimir Append); Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Imprimir → cadena Num char Append → + Imprimir Append); Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Imprimir → cadena Num char Append → + Imprimir Append); Lectura → PRLeer ValLeer; ValLeer → (TokId)			TokId
Append Himprimir Himprimir Append Himprimir A			
char	Imprimir	→	
Append → Hmprimir Append			
Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Lectura → PRLeer ValLeer; ValLeer → (TokId)			
Lectura → PRLeer ValLeer; ValLeer → (TokId)	A 1		- Imaginia Amand
Lectura → PRLeer ValLeer; ValLeer → (TokId)	Appena	\rightarrow	+ Imprimir Appena
ValLeer → (TokId) PRSI DeseCondicion Cond2			1);
ValLeer → (TokId) PRSI DeseCondicion Cond2			
ValLeer → (TokId) PRSI DeseCondicion Cond2			
ValLeer → (TokId) PRSI DeseCondicion Cond2			
ValLeer → (TokId) PRSI DeseCondicion Cond2			
ValLeer → (TokId) PRSI DeseCondicion Cond2	_		
DDSI DeseCondision Cond2	Lectura	\rightarrow	PRLeer ValLeer;
DDSI DeseCondicion Cond?			
DDSI DeseCondicion Cond?			
Condicion → PRSI DescCondicion Cond2	ValLeer	\rightarrow	(TokId)
Condicion → PRSI DescCondicion Cond2			
Condicion → PRSI DescCondicion Cond2			
Condicion	Condicion	_	PRSI DescCondicion Cond2
	Condicion		

DescCondicion	→	ValCondicion CuerpoDescCond
CuerpoDescCond	→	{ Texto
Cond2	→	PRSINO_SI DescCondicion Cond2 PRSINO CuerpoDescCond e
ValCondicion	→	(DescValCondicion)
DescValCondicion	→	SigDif Condicionante CondicionantePlus
Condicionante	→	Suma OpRel Suma PRBooleana
SigDif	→	! SigDif e
CondicionantePlus	→	OpLogico DescValCondicion
CicloM	→	PRMIENTRAS ValCondicion CuerpoDescCond
CicloHM	→	DescCicloHM ValCondicion
DescCicloHM	>	PRHACER CuerpoDescCond PRMIENTRAS
CicloF	\rightarrow	EstructCicloF CuerpoDescCond
EstructCicloF	\rightarrow	EstructDesde EstructHasta EstructIncremento

EstructDesde	\rightarrow	PRDESDE desde
EstructHasta	→	PRHASTA hasta
EstructIncremento	>	PRINCREMENTO NumE
desde	→	TokId = NumE
hasta	→	TokId OpRel NumE

Primeros:

No Terminal	Primeros
Codigo	PRprincipal
SimboloInicial	(
Texto	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
DeclaVariable	Tokld
Incrementable	++,
VariablesL	PRDatoPrimi
id	II II .
idVar	Tokld
valor	=, e
TipoValor	cadena, -, Tokid, Num, (, booleano, char
Suma	-, Tokld, Num, (
Suma'	+, -, e
Multi	-, Tokld, Num, (
Multi'	*,/,e
Pot	-, Tokld, Num, (
Pot'	^, e
Unar	-, Tokld, Num, (
Elem	Tokid, Num, (
Impresión	PRImprimir
ValImprimir	(
Imprimir	Tokld, cadena, Num, char
Append	+,)
Lectura	PRLeer
ValLeer	(
Condicion	PRSI
DescCondicion	(
CuerpoDescCond	{
Cond2	PRSINO_SI, PRSINO, e
ValCondicion	(
DescValCondicion	!, e
Condicionante	-, Tokld, Num, (
SigDif	!, e
CondicionantePlus	OpLogico, e
CicloM	PRMIENTRAS
CicloHM	PRHACER
DescCicloHM	PRHACER
CicloF	PRDESDE
EstructCicloF	PRDESDE
EstructDesde	PRDESDE
EstructHasta	PRHASTA
EstructIncremento	PRINCREMENTO
desde	Tokld
hasta	Tokld

Siguientes:

No Terminal	Siguientes
Codigo	\$
SimboloInicial	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
Texto	\$, PRSINO_SI, PRSINO, PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER,
	PRDESDE. }, PRMIENTRAS, e
DeclaVariable	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
Incrementable	;
VariablesL	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
id	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
idVar	II II .
valor	II II .
TipoValor	, II II .
Suma	;, ",", ;,), OpRel, OpLogico,), e
Suma'	;, ",", ;,), OpRel, OpLogico,), e
Multi	+, -, e, ';, ",", ;,), OpRel, OpLogico,), e
Multi'	+, -, e, ';, ",", ;,), OpRel, OpLogico,), e
Pot	*, /, e, +, -, e, ';, ",", ;,), OpRel, OpLogico,), e
Pot'	*, /, e, +, -, e, ';, ",", ;,), OpRel, OpLogico,), e
Unar	*, /, e, +, -, e, ';, ",", ;,), OpRel, OpLogico,), e
Elem	*, /, e, +, -, e, ';, ",", ;,), OpRel, OpLogico,), e
Impresión	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
Vallmprimir	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
Imprimir	+,)
Append	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
Lectura	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
ValLeer	;
Condicion	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
DescCondicion	PRSINO_SI, PRSINO, PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }, e
CuerpoDescCond	PRSINO_SI, PRSINO, PRDatoPrimi, Tokld, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }, PRMIENTRAS, e
Cond2	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
ValCondicion	{, PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
DescValCondicion)
Condicionante	OpLogico,), e
SigDif	-, Tokld, Num, (
CondicionantePlus)
CicloM	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
CicloHM	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
DescCicloHM	(
CicloF	PRDatoPrimi, TokId, PRImprimir, PRLeer, PRSI, PRMIENTRAS, PRHACER, PRDESDE. }
EstructCicloF	{
EstructDesde	PRHASTA
EstructHasta	PRINCREMENTO
EstructIncremento	{
desde	PRHASTA
hasta	PRINCREMENTO