



Tecnológico Nacional de México

Instituto Tecnológico de Veracruz

EJ23 6J1A - Lenguajes y Autómatas I

Docente: Ofelia Gutiérrez Giraldi

Proyecto: Fase Sintáctica del Compilador

Equipo 2 - Integrantes:

Honorio Acosta Ruiz

Laura Espejo Alvarado

Kitzia Guadalupe Munive Cabal

H. Veracruz, Ver., 18 de mayo 2023

Tabla de contenido

ELL (EasyLearningLanguage)	2
Propósito	2
Origen del nombre ELL: EasyLearningLanguage	2
Manual de Usuario	3
Conceptos Básicos	3
Tipo de Datos	3
Palabras Clave	4
Operadores	5
Sintaxis	5
Variables en ELL	5
Imprimir en ELL	6
Sentencias decisión en ELL	7
Sentencias Bucle en ELL	9
Tabla de errores	11
Manual del Sistema	13
Requisitos del sistema	13
Descripción del sistema	13
Tabla de valores	13
Conceptos Básicos	15
Tipo de Datos	15
Operadores	16
Sintaxis	16
Variables en ELL	16
Imprimir en ELL	17
Sentencias decisión en ELL	18
Sentencias Bucle en ELL	20
Gramáticas del Lenguaje	22
Tabla de errores.	24

ELL (EasyLearningLanguage)

Propósito

EasyLearningLanguage (ELL) se propone a ser un lenguaje de programación tipado y de propósito general que será diseñado para permitir el desarrollo de aplicaciones básicas.

Se centrará en la lógica de programación básica, lo que lo hará adecuado para la enseñanza de conceptos fundamentales como variables, operaciones, condicionales y ciclos. Aunque ELL no tendrá una amplia gama de características avanzadas, será diseñado para ser fácilmente escalable a medida que los usuarios adquieren más habilidades en programación.

ELL será creado con el propósito de ser un lenguaje de programación en español de nivel básico.

Origen del nombre ELL: EasyLearningLanguage



Figura 1 - Icono del lenguaje

El nombre de nuestro lenguaje es ELL, debido a que es un lenguaje de programación orientado a facilitar el aprendizaje de la lógica básica de programación para programadores novatos.

El nombre ELL proviene de las siglas EasyLearningLanguage, lo que en español significa Lenguaje Fácil de Aprender.

Manual de Usuario

El respectivo manual busca proporcionar una guía clara y completa para nuestro lenguaje que permita a los programadores utilizar el lenguaje de manera efectiva, donde como objetivos tiene los siguientes:

- Facilitar el aprendizaje del lenguaje:

El manual proporciona una introducción clara al lenguaje de programación, explicando los conceptos básicos, sintaxis y características únicas del lenguaje.

- Ayudar a los programadores a detectar y corregir errores:

Incluye información sobre los errores que pueden ocurrir durante la programación con el lenguaje, incluyendo como corregir estos errores.

- Proporcionar información de referencia:

Es una fuente completa de información de referencia para el lenguaje, que incluye una lista detallada de las palabras clave, operadores y tipos de datos; permitiendo a los programadores buscar rápidamente información específica cuando la necesita.

Conceptos Básicos

Tipo de Datos

El lenguaje ELL tiene los siguientes tipos de datos básicos:

- **Entero:** Tipo de dato que representa un numero entero.
- **Flotante:** Tipo de dato que representa un numero en decimal.
- **Cadena:** Tipo de dato que representa un texto.
- **Carácter:** Tipo de datos que representa un símbolo.
- **Booleano:** Tipo de dato que representa aquellos que tienen un valor de verdadero o falso.

Palabras Clave

Las palabras clave del lenguaje ELL son:

Palaba clave	Descripción
Inicio	Indica el arranque del programa.
Fin	Indica el cierre del programa.
Establecer	Indica la definición de una variable.
Interpretar	Indica la lectura de una variable mediante el tipo de dato.
Escribir	Indica la escritura de una variable o literal en terminal.
Falso	Indica un valor lógico falso.
Verdadero	Indica un valor lógico verdadero.
Para	Indica el inicio de un ciclo Para.
Hasta que	Indica la condición a evaluar del ciclo Para.
Con incremento	Indica el incremento del ciclo Para (el incremento por defecto es 1).
FinPara	Indica el final del ciclo Para.
Mientras	Indica el inicio del ciclo Mientras.
FinMientras	Indica el final del ciclo Mientras.
Repetir	Indica el inicio del ciclo Repetir.
Finaliza cuando	Indica la condición a evaluar del ciclo Repetir.
Si	Indica el inicio del condicional Si.
Entonces	Indica el cuerpo del condicional Si.
Sino	Indica el caso contrario del condicional Si.
FinSi	Indica el final del condicional Si.
Conforme	Indica el inicio del condicional Conforme.
Hacer	Indica el cuerpo del condicional Conforme.
Caso	Indica un caso del condicional Conforme.
En otro caso	Indica el caso por defecto del condicional Conforme.
FinConforme	Indica el fin del condicional Conforme.

Tabla 1 - Palabras clave

Operadores

Los operadores con los que cuenta el lenguaje ELL son:

Operador	Descripción
=	Operador que indica asignación.
+	Operador que indica una suma.
-	Operador que indica una resta.
*	Operador que indica una multiplicación.
/	Operador que indica una división.
%	Operador que indica el módulo.
&&	Operador lógico AND.
	Operador lógico OR.
!	Operador lógico NOT.
==	Operador de comparación igual que.
!=	Operador de comparación diferente.
>	Operador de comparación mayor que.
<	Operador de comparación menor que.
>=	Operador de comparación mayor o igual que.
<=	Operador de comparación menor o igual que.

Tabla 2 - Operadores

Sintaxis

Variables en ELL

Las variables son un espacio de memoria para almacenar un dato. Para definir una variable seguimos esta estructura:

Inicio

Establecer [Tipo_De_Dato][Nombre_Variable];

Fin

Figura 2 - Estructura de cómo crear variables.

Algunos ejemplos de variables serían:

Inicio

Establecer Entero edad; # Esta es una variable entera

Establecer Flotante estatura; # Esta es una variable flotante

Establecer Cadena nombre; # Esta es una variable cadena

Establecer Carácter sexo; # Esta es una variable tipo carácter

Establecer Booleano estaVivo; # Esta es una variable tipo booleana

Fin

Figura 3 - Ejemplo de creación de variables.

Imprimir en ELL

Imprimir se refiere a mostrar información en la salida de texto del programa; es una forma fácil de visualizar resultados, mensajes o datos. Para imprimir definimos la siguiente estructura:

Inicio

Escribir "Cadena_de_Texto";

Escribir [variable];

Fin

Figura 4 - Estructura de Escribir

Solo se puede hacer una acción a la vez, es decir, si vas a imprimir una cadena solamente será la cadena, pero no puedes imprimir una cadena y una variable juntas.

Sentencias decisión en ELL

Las sentencias de decisión son sentencias que nos permiten tomar una decisión para poder ejecutar un bloque de sentencias u otro. Las sentencias de decisión que contiene el lenguaje son:

Si – Entonces – Sino

La estructura de la sentencia Si-Entonces-Sino es:

Inicio

Si [condición] Entonces

#Bloque de sentencias

Sino

#Bloque de sentencias

FinSi

Fin

Figura 5 - Estructura de Si-Entonces-Sino

La parte del Sino no tiene por qué existir. En este caso tendríamos una sentencia Si-Entonces.

Inicio

Si [condición] Entonces

#Bloque de sentencias

FinSi

Fin

Figura 6 - Sentencia Si - Entonces

La sentencia Si-Entonces-Sino pueden estar anidadas y así nos encontraríamos con una sentencia Si-Entonces-SinoSi, la cual tendría la siguiente estructura:

Inicio

Si [condición] Entonces

#Bloque de sentencias

Sino Si [condición] Entonces

#Bloque de sentencias

Sino Si [condición] Entonces

#Bloque de sentencias

Sino

#Bloque de sentencias

FinSi

Fin

Figura 7-Sentencia Si-Entonces-SinoSi

Conforme

Para los casos en los que se tienen muchas ramas o caminos de ejecución en una sentencia Si tenemos la sentencia Conforme. La sentencia Conforme evalúa una expresión y ejecutara el bloque de sentencias que coincida con el valor de la expresión.

El valor de la expresión puede ser numérico o al igual se pueden utilizar expresiones cuya evaluación sean cadenas.

La estructura de la sentencia Conforme es:

Inicio

Conforme [condición] Hacer

Caso [valor1]:

#Bloque de sentencias

Caso [valor2]:

#Bloque de sentencias

En Otro Caso:

#Bloque de sentencias

FinConforme

Fin

Figura 8 - Estructura de sentencia Conforme

Sentencias Bucle en ELL

Las sentencias de bucle nos van a permitir ejecutar un bloque de sentencias tantas veces como queramos, o tantas veces como se cumpla una condición. Las sentencias de bucle que contiene el lenguaje son:

Para

La estructura del bucle Para es:

Inicio

Para [sentencia_inicio] Hasta que [condición] Con incremento

[valor_entero_ó_decimal]

#Bloque de sentencias

FinPara

Fin

Figura 9-Estructura Para

Las funcionalidades en las que podemos utilizar la sentencia Para puede ser como un contador.

Mientras

La estructura repetitiva Mientras realiza una primera evaluación antes de ejecutar el bloque. Si la expresión es verdadera pasa a ejecutar de forma repetitiva el bloque de sentencias.

La estructura de la sentencia Mientras es la siguiente:

Inicio

Mientras [condición] Hacer

#Bloque de sentencias

FinMientras

Fin

Figura 10-Estructura de sentencia Mientras

Repetir

La estructura de la sentencia Repetir es la siguiente:

Inicio

Repetir

#Bloque de sentencias

Finalizar cuando [condición];

Fin

Figura 11-Estructura de sentencia Repetir

Tabla de errores

Número de error	Tipo	Ubicación	Token	Descripción	Solución.
1	Error léxico	Línea 3, columna 6	,	Símbolo ‘,’ desconocido	Remover símbolo ‘,’
2	Error sintáctico	Línea 3, Columna 24	;	Se encontró símbolo “;”	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - <CADENA_TEXTO> - <CARÁCTER_TEXTO> - <NUMERO_ENTERO> - <NUMERO_DECIMAL> - “Falso” - “Verdadero” - “!” - “(“ - <VARIABLE>
3	Error sintáctico	Línea 13, columna 32	Mientras	Se encontró el símbolo “Mientras”	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - “,” - “+” - “_” - “*” - “/” - “%” - “ ” - “==” - “!=” - “>” - “<” - “>=” - “<=”
4	Error sintáctico	Línea 19, columna 9	<VARIABLE>	Se encontró el símbolo <VARIABLE>	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - <CADENA_TEXTO> - <CARÁCTER_TEXTO> - <NUMERO_ENTERO> - <NUMERO_DECIMAL> - “Falso” - “Verdadero”
5	Error sintáctico	Línea 1, columna 1	Fin	Se encontró el símbolo “Fin”	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - <COMENTARIO> - “Interpretar” - “Escribir”

					<ul style="list-style-type: none">- "Establecer"- "Para"- "Mientras"- "Repetir"- "Si"- "Conforme"- <VARIABLE>
--	--	--	--	--	---

Manual del Sistema

El respectivo manual busca proporcionar una guía clara y completa para nuestro lenguaje que permita a los programadores utilizar el lenguaje de manera efectiva y dar información clara del lenguaje, donde como objetivos tiene los siguientes:

- Describir el sistema: El manual de sistema tiene como objetivo principal proporcionar una descripción completa y detallada del sistema.
- Orientar en la solución de problemas: El manual debe proporcionar información sobre la solución de problemas comunes que los usuarios pueden enfrentar al utilizar el sistema.
- Servir como referencia: El manual de sistema debe ser una referencia útil para los usuarios, brindando información detallada sobre todas las funciones, configuraciones y aspectos técnicos del sistema.

Requisitos del sistema

- Tener Java instalado.
- Tener JavaCC instalado.

Descripción del sistema

EC (EasyCompiler) es un compilador de código abierto que admite el lenguaje de programación ELL (EasyLearningLanguage). El compilador se ejecuta en cualquier plataforma que tenga instalada Java.

Tabla de valores

Valor	Tipo	Expresión regular
Inicio	PR_Arranque_Programa	Inicio
Fin	PR_Cierre_Programa	Fin
#	Simbolo_Comentario	#{letra símbolo número)*
edad	Variable	(letra _)(número letra _)*
Establecer	PR_Definicion_Variable	Establecer
Entero	PR_Tipo_De_Dato	Entero entero

Flotante	PR_Tipo_De_Dato	Flotante flotante
Cadena	PR_Tipo_De_Dato	Cadena cadena
Carácter	PR_Tipo_De_Dato	Character caracter
Booleano	PR_Tipo_De_Dato	Booleano booleano
;	Delimitador	;
Interpretar	PR_Lectura	Interpretar
Escribir	PR_Escritura	Escribir
123	Numero_Entero	$(- \epsilon +)(\text{número})^+$
123.123	Numero_Flotante	$(- \epsilon +)(\text{número})^* . (\text{número})^+$
"Cadena"	Cadena_Texto	" (letra número símbolo) * "
'C'	Caracter_Texto	' (letra número símbolo ϵ) '
=	Operador_Asignacion	=
+	Operador_Suma	+
-	Operador_Resta	-
*	Operador_Mult	*
/	Operador_Div	/
%	Operador_Mod	%
Falso	PR_Booleano_Falso	Falso
Verdadero	PR_Booleano_Verdadero	Verdadero
&&	Operador_Logico_AND	&&
	Operador_Logico_OR	
!	Operador_Logico_NOT	!
==	Operador_igualQue	==
!=	Operador_diferente	!=
>	Operador_mayorQue	>
<	Operador_menorQue	<
>=	Operador_mayorIgualQue	>=
<=	Operador_menorIgualQue	<=
Para	PR_Ciclo_Para	Para
Hasta que	PR_Condicion_Ciclo_Para	Hasta que

Con incremento	PR_Incremento_Ciclo_Para	Con incremento
FinPara	PR_Fin_Ciclo_Para	FinPara
Mientras	PR_Ciclo_Mientras	Mientras
FinMientras	PR_Fin_Ciclo_Mientras	FinMientras
Repetir	PR_Ciclo_Repetir	Repetir
Finaliza cuando	PR_Condicion_Ciclo_Repetir	Finaliza cuando
Si	PR_Condicional_Si	Si
Entonces	PR_Entonces	Entonces
Sino	PR_Conficional_Sino	Sino
FinSi	PR_Fin_Condicional_Si	FinSi
Conforme	PR_Condicional_Conforme	Conforme
Hacer	PR_Hacer	Hacer
Caso	PR_Conforme_Caso	Caso
:	Operador_Dos_Puntos	:
En otro caso	PR_Conforme_Caso_Predeterminado	En otro caso
FinConforme	PR_Fin_Condicional_Conforme	FinConforme
(Parentesis_Abierto	(
)	Parentesis_Cerrado)

Tabla 3 – Tabla de valores

Conceptos Básicos

Tipo de Datos

El lenguaje ELL tiene los siguientes tipos de datos básicos:

- **Entero:** Tipo de dato que representa un numero entero.
- **Flotante:** Tipo de dato que representa un numero en decimal.
- **Cadena:** Tipo de dato que representa un texto.
- **Carácter:** Tipo de datos que representa un símbolo.
- **Booleano:** Tipo de dato que representa aquellos que tienen un valor de verdadero o falso.

Operadores

Los operadores con los que cuenta el lenguaje ELL son:

Operador	Descripción
=	Operador que indica asignación.
+	Operador que indica una suma.
-	Operador que indica una resta.
*	Operador que indica una multiplicación.
/	Operador que indica una división.
%	Operador que indica el módulo.
&&	Operador lógico AND.
	Operador lógico OR.
!	Operador lógico NOT.
==	Operador de comparación igual que.
!=	Operador de comparación diferente.
>	Operador de comparación mayor que.
<	Operador de comparación menor que.
>=	Operador de comparación mayor o igual que.
<=	Operador de comparación menor o igual que.

Tabla 3 - Operadores

Sintaxis

Variables en ELL

Las variables son un espacio de memoria para almacenar un dato. Para definir una variable seguimos esta estructura:

Inicio

Establecer [Tipo_De_Dato][Nombre_Variable];

Fin

Figura 12 - Estructura de cómo crear variables.

Algunos ejemplos de variables serían:

Inicio

Establecer Entero edad; # Esta es una variable entera

Establecer Flotante estatura; # Esta es una variable flotante

Establecer Cadena nombre; # Esta es una variable cadena

Establecer Carácter sexo; # Esta es una variable tipo carácter

Establecer Booleano estaVivo; # Esta es una variable tipo booleana

Fin

Figura 13 - Ejemplo de creación de variables.

Imprimir en ELL

Imprimir se refiere a mostrar información en la salida de texto del programa; es una forma fácil de visualizar resultados, mensajes o datos. Para imprimir definimos la siguiente estructura:

Inicio

Escribir "Cadena_de_Texto";

Escribir [variable];

Fin

Figura 14 - Estructura de Escribir

Solo se puede hacer una acción a la vez, es decir, si vas a imprimir una cadena solamente será la cadena, pero no puedes imprimir una cadena y una variable juntas.

Sentencias decisión en ELL

Las sentencias de decisión son sentencias que nos permiten tomar una decisión para poder ejecutar un bloque de sentencias u otro. Las sentencias de decisión que contiene el lenguaje son:

Si – Entonces – Sino

La estructura de la sentencia Si-Entonces-Sino es:

Inicio

Si [condición] Entonces

#Bloque de sentencias

Sino

#Bloque de sentencias

FinSi

Fin

Figura 15 - Estructura de Si-Entonces-Sino

La parte del Sino no tiene por qué existir. En este caso tendríamos una sentencia Si-Entonces.

Inicio

Si [condición] Entonces

#Bloque de sentencias

FinSi

Fin

Figura 16 - Sentencia Si - Entonces

La sentencia Si-Entonces-Sino pueden estar anidadas y así nos encontraríamos con una sentencia Si-Entonces-SinoSi, la cual tendría la siguiente estructura:

Inicio

Si [condición] Entonces

#Bloque de sentencias

Sino Si [condición] Entonces

#Bloque de sentencias

Sino Si [condición] Entonces

#Bloque de sentencias

Sino

#Bloque de sentencias

FinSi

Fin

Figura 17-Sentencia Si-Entonces-SinoSi

Conforme

Para los casos en los que se tienen muchas ramas o caminos de ejecución en una sentencia Si tenemos la sentencia Conforme. La sentencia Conforme evalúa una expresión y ejecutara el bloque de sentencias que coincida con el valor de la expresión.

El valor de la expresión puede ser numérico o al igual se pueden utilizar expresiones cuya evaluación sean cadenas.

La estructura de la sentencia Conforme es:

Inicio

Conforme [condición] Hacer

Caso [valor1]:

#Bloque de sentencias

Caso [valor2]:

#Bloque de sentencias

En Otro Caso:

#Bloque de sentencias

FinConforme

Fin

Figura 18 - Estructura de sentencia Conforme

Sentencias Bucle en ELL

Las sentencias de bucle nos van a permitir ejecutar un bloque de sentencias tantas veces como queramos, o tantas veces como se cumpla una condición. Las sentencias de bucle que contiene el lenguaje son:

Para

La estructura del bucle Para es:

Inicio

Para [sentencia_inicio] Hasta que [condición] Con incremento

[valor_entero_ó_decimal]

#Bloque de sentencias

FinPara

Fin

Figura 19-Estructura Para

Las funcionalidades en las que podemos utilizar la sentencia Para puede ser como un contador.

Mientras

La estructura repetitiva Mientras realiza una primera evaluación antes de ejecutar el bloque. Si la expresión es verdadera pasa a ejecutar de forma repetitiva el bloque de sentencias.

La estructura de la sentencia Mientras es la siguiente:

```
Inicio  
  
    Mientras [condición] Hacer  
  
        #Bloque de sentencias  
  
    FinMientras  
  
Fin
```

Figura 20-Estructura de sentencia Mientras

Repetir

La estructura de la sentencia Repetir es la siguiente:

```
Inicio  
  
    Repetir  
  
        #Bloque de sentencias  
  
    Finalizar cuando [condición];  
  
Fin
```

Figura 21-Estructura de sentencia Repetir

Gramáticas del Lenguaje

$\langle \text{PROGRAMA} \rangle \rightarrow \text{Inicio (} \langle \text{SENTENCIAS} \rangle \text{)}^+ \text{Fin}$

$\langle \text{CONSTANTES} \rangle \rightarrow \text{NUMERO_ENTERO} \mid \text{NUMERO_DECIMAL} \mid \text{CADENA_TEXTO} \mid$
 $\text{CARACTER_TEXTO} \mid \text{BOOLEANO_FALSO} \mid \text{BOOLEANO_VERDADERO}$

$\langle \text{TIPO_DATO} \rangle \rightarrow \text{ENTERO} \mid \text{FLOTANTE} \mid \text{CADENA} \mid \text{CARACTER} \mid \text{BOOLEANO}$

$\langle \text{OPERADORES_RELACIONALES} \rangle \rightarrow == \mid != \mid > \mid >= \mid < \mid <=$

$\langle \text{OPERADORES_ARITMETICOS} \rangle \rightarrow + \mid - \mid * \mid / \mid \%$

$\langle \text{OPERADORES_LOGICOS} \rangle \rightarrow \&\& \mid \text{"||"}$

$\langle \text{OPERADORES} \rangle \rightarrow \langle \text{OPERADORES_ARITMETICOS} \rangle \mid \langle \text{OPERADORES_LOGICOS} \rangle$

$\langle \text{ASIGNACION} \rangle \rightarrow (= \mid \text{ASIGNACION_COMPUESTA}) \langle \text{CONDICION} \rangle$

$\langle \text{CONDICION} \rangle \rightarrow \langle \text{OPERACION} \rangle (\langle \text{OPERADORES_RELACIONALES} \rangle \langle \text{OPERACION} \rangle)^*$

$\langle \text{OPERACION} \rangle \rightarrow (!)^*(\text{VARIABLE} \mid \langle \text{CONSTANTES} \rangle \mid \langle \text{OPERACION_PARENTESIS} \rangle)$
 $(\langle \text{OPERADORES} \rangle (!)^*(\text{VARIABLE} \mid \langle \text{CONSTANTES} \rangle \mid \langle \text{OPERACION_PARENTESIS} \rangle))^*$

$\langle \text{OPERACION_PARENTESIS} \rangle \rightarrow "(" \langle \text{OPERACION} \rangle ")"$

$\langle \text{SENTENCIAS} \rangle \rightarrow \text{COMENTARIO}$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_ASIGNACION} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_DECLARACION_VARIABLE} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{LEER_DATO} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{IMPRIMIR_DATO} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_SI} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_CONFORME} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_PARA} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_REPETIR} \rangle$

$\langle \text{SENTENCIAS} \rangle \rightarrow \langle \text{SENTENCIA_MIENTRAS} \rangle$

$\langle \text{DECLARACION_VARIABLE} \rangle \rightarrow \text{Establecer } \langle \text{TIPO_DATO} \rangle \text{ VARIABLE } (\langle \text{ASIGNACION} \rangle)?$

$\langle \text{SENTENCIA_ASIGNACION} \rangle \rightarrow \text{VARIABLE } \langle \text{ASIGNACION} \rangle ;$

$\langle \text{SENTENCIA_DECLARACION_VARIABLE} \rangle \rightarrow \langle \text{DECLARACION_VARIABLE} \rangle ;$

$\langle \text{LEER_DATO} \rangle \rightarrow \text{Interpretar VARIABLE} ;$

$\langle \text{IMPRIMIR_DATO} \rangle \rightarrow \text{Escribir } (\langle \text{CONSTANTES} \rangle \mid \text{VARIABLE}) ;$

$\langle \text{SENTENCIA_SI} \rangle \rightarrow \text{Si } \langle \text{CONDICION} \rangle \text{ Hacer } (\langle \text{SENTENCIAS} \rangle)^+$

$(\text{Sino } (\langle \text{SENTENCIAS} \rangle)^+)? \text{ FinSi}$

$\langle \text{SENTENCIA_CONFORME} \rangle \rightarrow \text{Conforme VARIABLE Hacer}$

$(\text{Caso } \langle \text{CONSTANTES} \rangle : (\langle \text{SENTENCIAS} \rangle)^+)^+$

$(\text{En otro caso} : (\langle \text{SENTENCIAS} \rangle)^+)? \text{ FinConforme}$

$\langle \text{SENTENCIA_PARA} \rangle \rightarrow \text{Para } (\text{VARIABLE} = (\langle \text{CONSTANTES} \rangle \mid \text{VARIABLE}) \mid \langle \text{DECLARACION_VARIABLE} \rangle)$

$\text{Hasta que } \langle \text{CONDICION} \rangle \text{ Con incremento } (\text{NUMERO_ENTERO} \mid \text{NUMERO_DECIMAL})$

$(\langle \text{SENTENCIAS} \rangle)^+ \text{ FinPara}$

$\langle \text{SENTENCIA_REPETIR} \rangle \rightarrow \text{Repetir } (\langle \text{SENTENCIAS} \rangle)^+ \text{ Finaliza cuando } \langle \text{CONDICION} \rangle ;$

$\langle \text{SENTENCIA_MIENTRAS} \rangle \rightarrow \text{Mientras } \langle \text{CONDICION} \rangle \text{ Hacer } (\langle \text{SENTENCIAS} \rangle)^+ \text{ FinMientras}$

Tabla de errores.

Número de error	Tipo	Ubicación	Token	Descripción	Solución.
1	Error léxico	Línea 3, columna 6	,	Símbolo ‘,’ desconocido	Remover símbolo ‘,’
2	Error sintáctico	Línea 3, Columna 24	;	Se encontró símbolo “;”	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - <CADENA_TEXTO> - <CARÁCTER_TEXTO> - <NUMERO_ENTERO> - <NUMERO_DECIMAL> - “Falso” - “Verdadero” - “!” - “(“ - <VARIABLE>
3	Error sintáctico	Línea 13, columna 32	Mientras	Se encontró el símbolo “Mientras”	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - “,” - “+” - “_” - “*” - “/” - “%” - “ ” - “==” - “!=” - “>” - “<” - “>=” - “<=”
4	Error sintáctico	Línea 19, columna 9	<VARIABLE>	Se encontró el símbolo <VARIABLE>	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - <CADENA_TEXTO> - <CARÁCTER_TEXTO> - <NUMERO_ENTERO> - <NUMERO_DECIMAL> - “Falso” - “Verdadero”
5	Error sintáctico	Línea 1, columna 1	Fin	Se encontró el símbolo “Fin”	Se esperaba uno de los siguientes: <ul style="list-style-type: none"> - <COMENTARIO> - “Interpretar” - “Escribir”

					<ul style="list-style-type: none">- "Establecer"- "Para"- "Mientras"- "Repetir"- "Si"- "Conforme"- <VARIABLE>
--	--	--	--	--	---