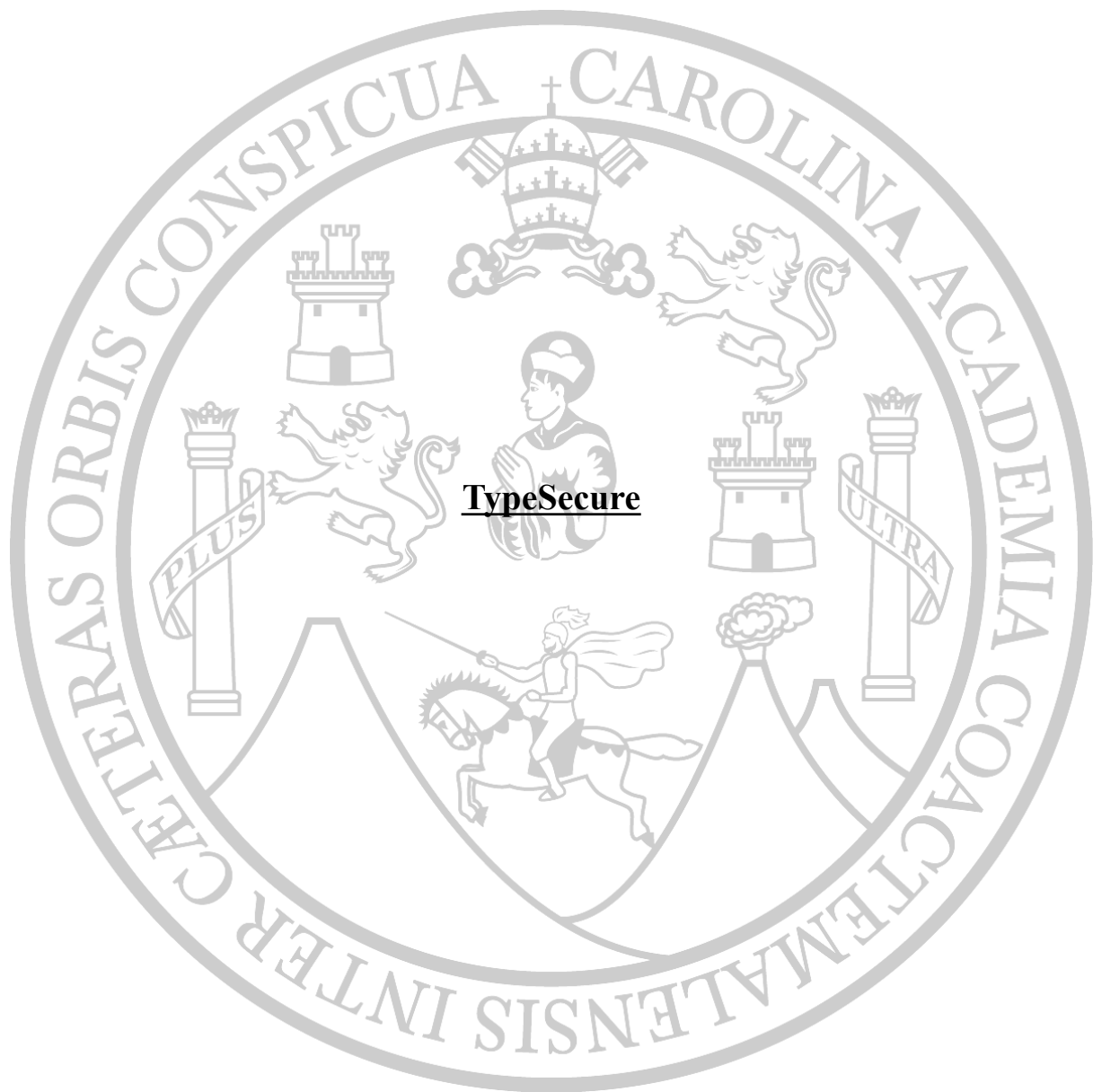


Universidad de San Carlos de Guatemala
Centro Universitario de Occidente
División Ciencias de la Ingeniería
Organización de lenguajes y compiladores 1



Nombre:
Pedro Ricardo Gordillo González

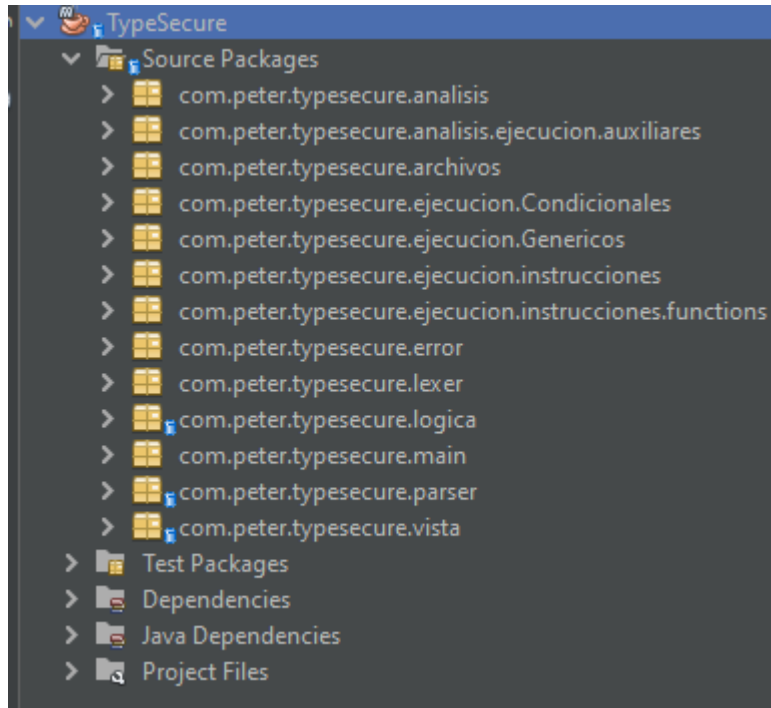
Registro Académico:
202031683

Quetzaltenango, mayo de 2023

Manual de usuario

Organización del proyecto

El proyecto se organiza de la siguiente manera.



El paquete `com.peter.typesecure.analisis` contiene la clase que se encarga de realizar el análisis léxico y sintáctico por medio de las herramientas de jflex y cup.

El paquete `com.peter.typesecure.analisis.ejecucion.auxiliares` contiene las clases `Function`, `Parameter` y `VariableAux`, estas se encargan de crear una función y un parámetro respectivamente.

El paquete `com.peter.typesecure.archivos` contiene la clase que se encarga de generar las instrucciones para generar el ast y los archivos html.

El paquete `com.peter.typesecure.ejecucion.Condicionales` contiene las clases sobre las cuales se basan las instrucciones `if`, `else` y `else if`.

El paquete `com.peter.typesecure.ejecucion.Genericos` contiene la clase padre `Instruction`, de esta clase heredan todas las instrucciones del lenguaje TypeSecure, así como las operaciones que pueden realizarse.

El paquete `com.peter.typesecure.ejecucion.instrucciones` contiene las clases que heredan de `Instruction` y realizan las ejecuciones de las instrucciones del lenguaje TypeSecure.

El paquete `com.peter.typesecure.ejecucion.instrucciones.functions` contiene las clases de los diferentes tipos de funciones que pueden declararse.

El paquete `com.peter.typesecure.error` contiene la clase que tiene la información de los errores hallados en los analizadores.

El paquete `com.peter.typesecure.lexer` contiene la clase generada por Jflex, esta contiene el analizador léxico del lenguaje TypeSecure.

El paquete `com.peter.typesecure.logica` realiza la ejecución de cada instrucción hallada.

El paquete `com.peter.typesecure.Main` contiene la clase Main, la cual ejecuta el programa.

El paquete `com.peter.typesecure.parser` contiene las clases generadas por Cup, esta contiene el analizador sintáctico del lenguaje TypeSecure.

El paquete `com.peter.typesecure.vista` contiene la interfaz gráfica de la aplicación.

Expresiones regulares

Inicio de un texto

`comillaD = \"`
`comillaS = '`

Id

`id = ([a-zA-z] | "_") ([a-zA-z] | {entero} | "_") *`

Espacios

`WhiteSpace = [\r\n|\r\n] | [\t\f]`

Números

`entero = [0-9] +`
`decimal = ({entero}) + (".") ({entero}) +`
`bigInt = {entero} "n"`

Booleanos

true = "true"
false = "false"

Palabras reservadas

Number = "number"
BigInt = "bigint"
String = "string"
Boolean = "boolean"
Void = "void"
Const = "const"
Let = "let"

Gramática

Esta produccion contiene la gramatica del lenguaje TypeSecure

inicio ::= ts

Esta produccion contiene la gramatica de las instrucciones del lenguaje TypeSecure

ts ::= instrucciones

Esta produccion detecta una o más instrucciones

instrucciones ::= instrucciones opciones_instrucciones
| opciones_instrucciones

Esta produccion detecta una instruccion, en esta opcion contiene la instruccion 'return'

opciones_instrucciones ::= declaracion_variable
| asignacion
| funcion_console_log
| declaracion_funcion
| llamada_funcion_normal
| opciones_if
| funcion_return
| instruccion_break
| instruccion_continue
| funcion_incremento_decremento
| funcion_for
| funcion_while
| funcion_do_while

- | funcion_printAst
- | funcion_getSymbolTable

Esta produccion detecta una instruccion, en esta opcion no se acepta la instruccion 'return'

opciones_instrucciones_sin_return ::=

- declaracion_variable
- | asignacion
- | funcion_console_log
- | declaracion_funcion
- | llamada_funcion_normal
- | opciones_if
- | instruccion_break
- | instruccion_continue
- | funcion_incremento_decremento
- | funcion_for
- | funcion_while
- | funcion_do_while
- | funcion_printAst
- | funcion_getSymbolTable

Esta produccion detecta una asignacion

asignacion ::= ID ASIGNACION variable_valor PUNTO_COMA

Esta produccion detecta una declaracion de una variable

declaracion_variable ::= tipo_acceso_variable id_valor_variable_lista PUNTO_COMA

Esta produccion detecta un conjunto de declaraciones de variables

id_valor_variable_lista ::= id_valor_variable_lista COMA id_valor_variable
| id_valor_variable

Esta produccion detecta el id de la variable, el tipo de la variable y el valor de la variable

id_valor_variable ::= ID DOS_PUNTOS tipo_dato variable ASIGNACION variable_valor
| ID:id ASIGNACION variable_valor
| ID DOS_PUNTOS tipo_dato variable

Esta produccion detecta el tipo de acceso de la variable a declarar

tipo_acceso_variable ::= RCONST
| RLET

Esta produccion detecta el tipo de dato de la variable a declarar

tipo_dato_variable ::= RNUMBER
| RBIGINT
| RSTRING
| RBOOLEAN

Esta produccion detecta el valor a asignar de una variable
variable_valor ::= instruccion

Esta produccion detecta el valor como tal de una variable
instruccion ::= operaciones_matematicas
 | operaciones_logicas
 | funcion_boolean

Esta producción detecta una operación matemática, un id, un texto, y deriva a las funciones numéricas estáticas y funciones estéticas del lenguaje

operaciones_matematicas ::=
 MENOS operaciones_matematicas
 | MAS operaciones_matematicas
 | operaciones_matematicas MAS operaciones_matematicas
 | operaciones_matematicas MENOS operaciones_matematicas
 | operaciones_matematicas MULTIPLICACION operaciones_matematicas
 | operaciones_matematicas DIVISION operaciones_matematicas
 | operaciones_matematicas MOD operaciones_matematicas
 | PARENTESIS_ABIERTO operaciones_matematicas
PARENTESIS_CERRADO
 | funcion_number
 | funcion_bigint
 | funcion_length
 | funciones_numericas_estaticas
 | NUMBER
 | BIGINT
 | ID
 | TEXTO
 | ID PARENTESIS_ABIERTO PARENTESIS_CERRADO
 | ID PARENTESIS_ABIERTO parametros_funcion_asignacion
PARENTESIS_CERRADO
 | variables_string
 | incremento
 | decremento

Esta producción detecta el nombre de las funciones numéricas estáticas

funciones_numericas_estaticas ::= MATH PUNTO ABS PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
 | MATH PUNTO CEIL PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO

```

| MATH PUNTO COS PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO SIN PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO TAN PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO EXP PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO FLOOR PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO POW PARENTESIS_ABIERTO
operaciones_matematicas COMA operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO SQRT PARENTESIS_ABIERTO
operaciones_matematicas PARENTESIS_CERRADO
| MATH PUNTO RANDOM PARENTESIS_ABIERTO
PARENTESIS_CERRADO
| MATH PUNTO E
| MATH PUNTO PI
| MATH PUNTO SQRT2

```

Esta produccion detecta la accion de incremento de una variable como asignación

```

incremento ::= ID INCREMENTO

```

Esta produccion detecta la accion de decremento de una variable como asignacion

```

decremento ::= ID DECREMENTO

```

Esta produccion detecta la accion de incremento o decremento de una variable como funcion

```

funcion_incremento_decremento ::=

```

```

    ID INCREMENTO PUNTO_COMA
| ID DECREMENTO PUNTO_COMA

```

Esta produccion las operaciones relacionales que pueden realizarse

```

operaciones_relacionales ::= operaciones_matematicas MAYOR_QUE

```

```

operaciones_matematicas

```

```

| operaciones_matematicas MAYOR_IGUAL operaciones_matematicas
| operaciones_matematicas MENOR_QUE operaciones_matematicas
| operaciones_matematicas MENOR_IGUAL operaciones_matematicas
| operaciones_matematicas COMPARACION operaciones_matematicas
| operaciones_matematicas DISTINTO operaciones_matematicas
| TRUE COMPARACION TRUE
| TRUE COMPARACION FALSE
| FALSE COMPARACION TRUE
| FALSE COMPARACION FALSE

```

```

| operaciones_matematicas COMPARACION TRUE
| TRUE COMPARACION operaciones_matematicas
| FALSE COMPARACION operaciones_matematicas
| TRUE DISTINTO TRUE
| TRUE DISTINTO FALSE
| FALSE DISTINTO TRUE
| FALSE DISTINTO FALSE
| operaciones_matematicas DISTINTO TRUE
| TRUE DISTINTO operaciones_matematicas
| operaciones_matematicas DISTINTO FALSE
| FALSE DISTINTO operaciones_matematicas
| TRUE
| FALSE
| funcion_boolean COMPARACION TRUE
| funcion_boolean COMPARACION FALSE
| TRUE COMPARACION funcion_boolean
| FALSE COMPARACION funcion_boolean
| funcion_boolean DISTINTO TRUE
| funcion_boolean DISTINTO FALSE
| TRUE DISTINTO funcion_boolean
| FALSE DISTINTO funcion_boolean
| funcion_boolean COMPARACION funcion_boolean
| funcion_boolean DISTINTO funcion_boolean
| PARENTESIS_ABIERTO operaciones_relacionales

```

PARENTESIS_CERRADO

Esta producción las operaciones lógicas que pueden realizarse

`operaciones_logicas ::= operaciones_logicas AND operaciones_logicas`

```

| operaciones_logicas OR operaciones_logicas
| NOT operaciones_logicas
| NOT operaciones_matematicas
| PARENTESIS_ABIERTO operaciones_logicas PARENTESIS_CERRADO
| operaciones_relacionales
| funcion_boolean AND funcion_boolean
| funcion_boolean OR funcion_boolean
| operaciones_logicas AND funcion_boolean
| funcion_boolean AND operaciones_logicas
| operaciones_logicas OR funcion_boolean
| funcion_boolean OR operaciones_logicas
| operaciones_matematicas OR operaciones_logicas
| operaciones_matematicas AND operaciones_logicas
| NOT funcion_boolean

```


Esta producción detecta la función length

funcion_length ::= operaciones_matematicas PUNTO LENGTH

Esta producción detecta la función number, que dentro de paréntesis puede recibir una operación matemática u operación lógica

funcion_number ::= FNUMBER PARENTESIS_ABIERTO operaciones_matematicas
PARENTESIS_CERRADO
| FNUMBER PARENTESIS_ABIERTO operaciones_logicas
PARENTESIS_CERRADO

Esta producción detecta la función bigint, que dentro de paréntesis puede recibir una operación matemática u operación lógica

funcion_bigint ::= FBIGINT PARENTESIS_ABIERTO operaciones_matematicas
PARENTESIS_CERRADO
| FBIGINT PARENTESIS_ABIERTO operaciones_logicas
PARENTESIS_CERRADO

Esta producción detecta la función boolean, que dentro de paréntesis puede recibir una operación matemática u operación lógica

funcion_boolean ::= FBOOLEAN PARENTESIS_ABIERTO operaciones_matematicas
PARENTESIS_CERRADO
| FBOOLEAN PARENTESIS_ABIERTO operaciones_logicas
PARENTESIS_CERRADO

Esta producción detecta la función string, que dentro de paréntesis puede recibir una operación matemática u operación lógica

funcion_string ::= FSTRING PARENTESIS_ABIERTO operaciones_matematicas
PARENTESIS_CERRADO
| FSTRING PARENTESIS_ABIERTO operaciones_logicas
PARENTESIS_CERRADO

Esta producción detecta la función charAt, que dentro de paréntesis puede recibir una operación matemática u operación lógica

funcion_char_at ::= operaciones_matematicas PUNTO CHARAT
PARENTESIS_ABIERTO index_char_at PARENTESIS_CERRADO

Esta producción detecta el index de la función charAt, que dentro de paréntesis puede recibir una operación matemática u operación lógica

`index_char_at ::= operaciones_matematicas`

Esta producción detecta la función concat que dentro de los paréntesis recibe una instrucción

`funcion_concat ::= operaciones_matematicas PUNTO CONCAT PARENTESIS_ABIERTO
instruccion PARENTESIS_CERRADO`

Esta producción se encarga de derivar a las funciones que tienen valor de tipo string

`variables_string ::= funcion_string
 | funcion_char_at
 | funcion_to_lower_case
 | funcion_to_upper_case
 | funcion_concat`

Esta producción detecta la función toLowerCase

`funcion_to_lower_case ::= operaciones_matematicas PUNTO TOLOWERCASE
PARENTESIS_ABIERTO PARENTESIS_CERRADO`

Esta producción detecta la función toUpperCase

`funcion_to_upper_case ::= operaciones_matematicas PUNTO TOUPPERCASE
PARENTESIS_ABIERTO PARENTESIS_CERRADO`

Esta producción detecta la función console.log

`funcion_console_log ::= CONSOLE PUNTO LOG PARENTESIS_ABIERTO
lista_instrucciones_console PARENTESIS_CERRADO PUNTO_COMA`

Esta producción deriva a las opciones que puede traer la instrucción console.log, puede ser una o más instrucciones

`lista_instrucciones_console ::= lista_instrucciones_console COMA instruccion
 | instruccion`

Esta producción detecta una declaración de una función

`declaracion_funcion ::=`

Esta producción detecta una declaración de una función de tipo void y sin parámetros

FUNCTION ID PARENTESIS_ABIERTO PARENTESIS_CERRADO DOS_PUNTOS
RVOID CORCHETE_ABIERTO instrucciones_funciones_sin_return
CORCHETE_CERRADO

Esta produccion detecta una declaracion de una funcion de tipo void y con parametros
|FUNCTION ID PARENTESIS_ABIERTO parametros_funcion_declaracion
PARENTESIS_CERRADO DOS_PUNTOS RVOID CORCHETE_ABIERTO
instrucciones_funciones_sin_return CORCHETE_CERRADO

Esta produccion detecta una declaracion de una funcion sin tipo y sin parametros
|FUNCTION ID PARENTESIS_ABIERTO PARENTESIS_CERRADO
CORCHETE_ABIERTO instrucciones_funcion CORCHETE_CERRADO

Esta produccion detecta una declaracion de una funcion sin tipo y con parametros
|FUNCTION ID PARENTESIS_ABIERTO parametros_funcion_declaracion
PARENTESIS_CERRADO CORCHETE_ABIERTO instrucciones_funcion
CORCHETE_CERRADO

Esta produccion detecta una declaracion de una funcion con tipo y sin parametros
|FUNCTION ID PARENTESIS_ABIERTO PARENTESIS_CERRADO DOS_PUNTOS
tipo_dato_variable CORCHETE_ABIERTO instrucciones_funcion
CORCHETE_CERRADO

Esta produccion detecta una declaracion de una funcion con tipo y con parametros
|FUNCTION ID PARENTESIS_ABIERTO parametros_funcion_declaracion
PARENTESIS_CERRADO DOS_PUNTOS tipo_dato_variable CORCHETE_ABIERTO
instrucciones_funcion CORCHETE_CERRADO

Esta produccion deriva a una o mas instrucciones sin return
instrucciones_funciones_sin_return ::= instrucciones_funciones_sin_return
opciones_instrucciones_sin_return
| opciones_instrucciones_sin_return

Esta produccion deriva a una o mas instrucciones con return
instrucciones_funcion ::= instrucciones

Esta produccion deriva a los parametros de las funciones, puede haber uno o mas parametros
parametros_funcion_declaracion ::= parametros_funcion_declaracion COMA
parametro_funcion_declaracion
| parametro_funcion_declaracion

Esta produccion deriva a los parametros de las funciones
parametro_funcion_declaracion ::= ID DOS_PUNTOS tipo_dato_variable

Esta produccion reconoce la instruccion return

funcion_return ::= RETURN instruccion PUNTO_COMA
| RETURN PUNTO_COMA

Esta produccion deriva a los parametros de las funciones, puede haber uno o mas parametros

parametros_funcion_asignacion ::= parametros_funcion_asignacion COMA instruccion
| instruccion

Esta produccion reconoce la invocacion de una funcion como instruccion, con o sin parametros

llamada_funcion_normal ::= ID PARENTESIS_ABIERTO PARENTESIS_CERRADO
PUNTO_COMA
| ID PARENTESIS_ABIERTO parametros_funcion_asignacion
PARENTESIS_CERRADO PUNTO_COMA

Esta produccion reconoce la instruccion if

funcion_if ::= IF PARENTESIS_ABIERTO instruccion PARENTESIS_CERRADO
CORCHETE_ABIERTO instrucciones_funcion CORCHETE_CERRADO

Esta produccion reconoce la instruccion else

funcion_else ::= ELSE CORCHETE_ABIERTO instrucciones_funcion
CORCHETE_CERRADO

Esta produccion reconoce la instruccion else if

funcion_else_if ::= ELSE IF PARENTESIS_ABIERTO instruccion
PARENTESIS_CERRADO CORCHETE_ABIERTO instrucciones_funcion
CORCHETE_CERRADO

Esta produccion reconoce un listado de if else

lista_else_if ::= lista_else_if funcion_else_if
| funcion_else_if

Esta produccion reconoce la instruccion if con sus derivaciones correspondientes

opciones_if ::= funcion_if
| funcion_if funcion_else
| funcion_if lista_else_if funcion_else
| funcion_if lista_else_if

Esta producción reconoce la instrucción break

instruccion_break ::= BREAK PUNTO_COMA

Esta produccion reconoce la instrucción continue

instruccion_continue ::= CONTINUE PUNTO_COMA

Esta producción reconoce la instrucción for

```
funcion_for ::=  
FOR PARENTESIS_ABIERTO declaracion_variable operaciones_logicas  
PUNTO_COMA accion_asignacion_for PARENTESIS_CERRADO  
CORCHETE_ABIERTO instrucciones_funcion CORCHETE_CERRADO  
| FOR PARENTESIS_ABIERTO asignacion operaciones_logicas PUNTO_COMA  
accion_asignacion_for PARENTESIS_CERRADO CORCHETE_ABIERTO  
instrucciones_funcion CORCHETE_CERRADO
```

Esta producción deriva a la asignación de la instrucción for

```
accion_asignacion_for ::= instruccion  
| ID ASIGNACION variable_valor  
| tipo_acceso_variable id_valor_variable_lista
```

Esta producción reconoce la instrucción while

```
funcion_while ::= WHILE PARENTESIS_ABIERTO instruccion  
PARENTESIS_CERRADO CORCHETE_ABIERTO instrucciones_funcion  
CORCHETE_CERRADO
```

Esta producción reconoce la instrucción do while

```
funcion_do_while ::= DO CORCHETE_ABIERTO instrucciones_funcion  
CORCHETE_CERRADO WHILE PARENTESIS_ABIERTO instruccion  
PARENTESIS_CERRADO PUNTO_COMA
```

Esta producción reconoce la función prinAst

```
funcion_printAst ::= PRINTAST PARENTESIS_ABIERTO instruccion  
PARENTESIS_CERRADO PUNTO_COMA
```

Esta producción reconoce la función getSymbolTable

```
funcion_getSymbolTable ::= GETSYMBOLTABLE PARENTESIS_ABIERTO  
PARENTESIS_CERRADO PUNTO_COMA
```