

Universidad Rafael Landivar  
Campus Quetzaltenango  
Compiladores

## PROYECTO

Gilda Maria Rivera Pac 1508217  
Jose Pablo Sanchez Estrada 1577817  
18 de noviembre de 2019

## FUNCIONAMIENTO BASICO DE COMPILADOR

Para poder utilizar el compilador, primero es necesario ingresar en el archivo números.txt el código en lenguaje loop que se querrá traducir al lenguaje ensamblador. Es importante que el código contenga la indentación adecuada, ya que si esta es de manera incorrecta, entonces podrían generarse errores o mal interpretarse el código. También es necesario que se cumpla con las especificaciones del lenguaje que se dan más. El código que se debe de ingresar puede ser como el del siguiente ejemplo:

```
Clase Impar
  Propiedades Publicas:
    entero a
    cadena b
  Metodos Publicos:
    cadena suma(entero a, cadena b)
      entero c = a + b
      devolver c
    cadena raizCuadrada(entero r)
      real root = raiz(r)
      devolver root

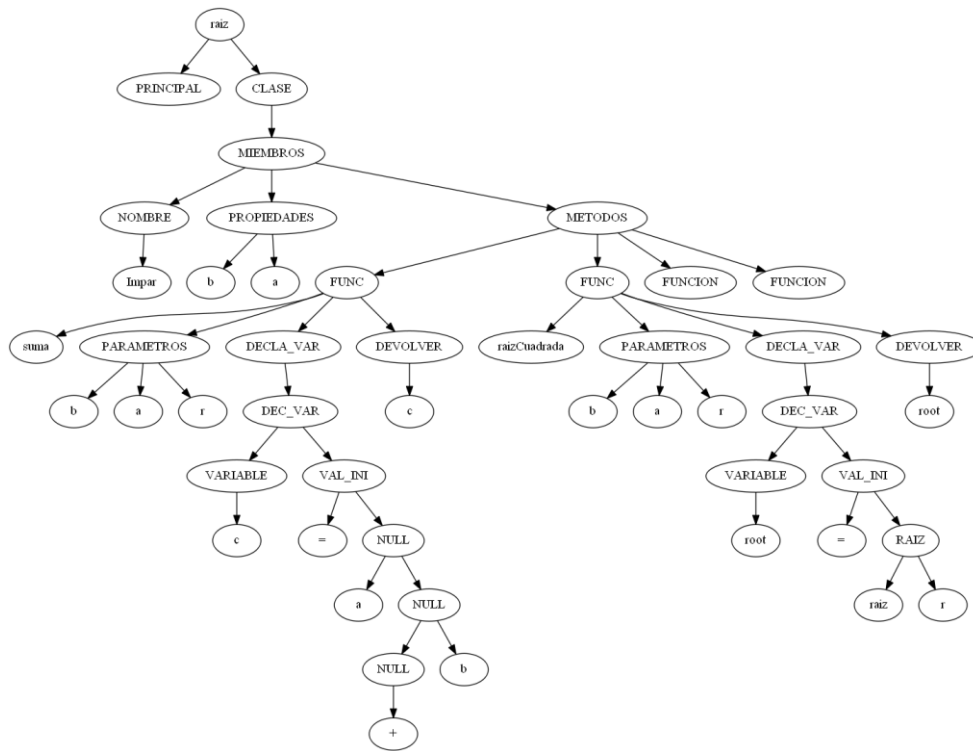
entero Principal()
  b = 5
  devolver 0
```

Después de haber ingresado el código, ya se es posible pasar al siguiente paso, el cual es ejecutar el programa, el código previamente ingresado pasará primero por el analizador léxico, y en ese momento se creará la tabla de símbolos, la cual estará en el archivo símbolos.txt; Esta tabla tiene los símbolos que contiene el código que se ha ingresado y la cantidad de veces que estos existen en este.

Luego de que se ha creado la tabla y se ha analizado el código con el analizador léxico, este pasa a ser revisado por el analizador sintáctico, el cual sigue la gramática cuyas reglas se presentan al final de este documento, este generará un árbol del código, el cual se llamará árbol.jpeg.

A continuación, se presenta el árbol y la tabla de símbolos que son generados usando el código ingresado en el ejemplo

## ARBOL



## TABLA SIMBOLOS

1	CLASE
1	VARIABLE CLASE
1	PROPIEDADES
1	METODOS
2	TIPO DE PROPIEDAD
2	DOS PUNTOS
1	COMA
1	SUMA
4	TIPO CADENA
1	TIPO REAL
1	RAIZ
5	TIPO ENTERO
1	PRINCIPAL
4	PARENTESIS1
4	PARENTESIS2
15	VARIABLE
3	IGUAL
3	DEVOLVER
2	NUMEROS DECIMALES

## **FUNCIONES SUELTAS**

Estas no pertenecen a ninguna clase, sino que son siempre públicas, cuando se utilizan es necesario declarar su tipo, darles un nombre, y si se desea pasarles parámetros. Los nombres de estas deben de seguir las reglas de las variables. Estas pueden devolver un valor que concuerde con el tipo de esta y sus argumentos son opcionales

## **FUNCIONES ESPECIALES**

Estas están predefinidas

- **CadenaAEntero**  
Esta recibe una cadena y la convierte en un entero, debe de ser un numero el contenido de esta
- **CadenaAReal**  
Esta recibe una cadena y la convierte a un número real, si no se tiene ningún numero se genera un error
- **CadenaABooleano**  
Convierte una cadena a booleano, se generará un error si esta cadena contiene un numero y se verifica si es verdadero o falso, de lo contrario no se puede convertir

## **FUNCIONES MATEMATICAS**

- **Seno**  
devuelve el cálculo del seno de un número que se envíe
- **Coseno**  
devuelve el cálculo del seno de un número que se envíe
- **Tangente**  
devuelve el cálculo de la tangente de un número que se envíe
- **Logaritmo**  
devuelve el cálculo de un logaritmo de base 10 de un número que se envíe
- **Raíz**  
devuelve el cálculo de raíz cuadrada de un número que se envíe

## **ESTRUCTURA ITERATIVA**

Permite ejecutar una o varias instrucciones, un número determinado de veces o indefinidamente, mientras se cumpla una condición, en este lenguaje se pueden utilizar dos tipos de estas, los cuales son:

- **Tipo desde:**

Estas deben de comenzar con la palabra desde, seguidamente puede haber una variable o una inicialización de variable a un valor, luego debe de ir la palabra mientras, seguidamente la condición ; es obligatorio que se incluya un incremento o decremento de un valor, para tener un límite de ciclo, por último, se incluye la palabra hacer y seguidamente las instrucciones que se realizarán mientras se cumpla la condición

- Tipo hacer:  
Estas deben de comenzar con la palabra hacer, seguidamente deben de ir las instrucciones que se desean; la palabra mientras y una condición, si esta se cumple se realiza el ciclo

## CLASES

Hay dos formas de declarar una clase en el lenguaje, una es instanciando una clase ya existente y la otra solo nombrándola como se haría con una función. El nombre de una clase debe de comenzar con una mayúscula, de lo contrario no se tomará como una clase. Cada clase puede tener sus propiedades y métodos, las propiedades deberán de ir declaradas antes que los métodos, estas también pueden tener constructores y destructores

- Propiedades  
Solo pueden declararse una vez, estas tienen tres distintos tipos de acceso, los cuales son:
  1. Publicas: Acceso público, se puede acceder desde cualquier punto
  2. Privadas: Acceso privado, solo se puede acceder desde los métodos de la clase
  3. Protegidas: Acceso solo a métodos de clase o a clases que hereden
- Métodos  
Solo pueden declararse una vez y deben de ir después que las propiedades, estos tienen tres distintos tipos de acceso, los cuales son:
  1. Públicos: Acceso público, se puede acceder desde cualquier punto
  2. Privados: Acceso privado, solo se puede acceder desde los métodos de la clase
  3. Protegidos: Acceso solo a métodos de clase o a clases que hereden
- Constructores  
Estos permiten asignar valores a las propiedades de las clases, estos se ejecutan al crear una instancia, puede recibir parámetros, pero no devolverá ningún valor, no son obligatorios
- Destructores  
Se ejecuta cuando una clase está siendo eliminada de la memoria, este no recibe parámetros y no devuelve ningún valor, no son obligatorios

## **INSTANCIACION**

Todas las clases que se hayan declarado pueden ser instanciadas por medio de la palabra reservada `instanciar`, y mediante un punto se accede a sus propiedades y métodos. Una variable del tipo de la clase debe de recibir la instancia

## **HERENCIA**

Se puede llevar a cabo utilizando la palabra `extiende`, por medio de esta se pueden extender funcionalidades de otra clase, es decir se obtienen métodos de la clase padre y estos pueden modificarse si se desea, siempre y cuando sean protegidas o publicas

## **FUNCION PRINCIPAL**

Esta es la función que se llama al iniciar la ejecución de un programa, esta debe de ir después que la declaración de todas las clases, esta puede recibir parámetros o no y siempre debe de devolver algún valor, este puede utilizar funciones sueltas que estén declaradas en cualquier parte. Es importante recalcar que debe de ir al final, de lo contrario no se reconoce como función clase

## **FIN DE LINEA:**

Cada línea de código no necesita un identificador de fin de línea, y tampoco utiliza un punto y coma al final

## **ESTRUCTURA SELECTIVA**

Esta ejecuta la condición, si es verdadera realiza una acción y si es falsa realiza otra acción, pueden realizarse condiciones dentro de condiciones

Condición:

- Mayor
- Menor
- Igual
- Distinto que

Siempre se devuelve un resultado booleano, estos resultados pueden asignarse a una variable, pueden anidarse varias estructuras

## **IDENTACION:**

Se debe de indentar correctamente para conocer que instrucciones forman parte de un bloque, y tomarse estas como parte de este, esto es esencial para el buen funcionamiento del programa

## **METODOS E/S**

## **LEER:**

Solo puede leerse una variable por vez, no es posible lo siguiente: *leer a,m* solo es posible leer una variable

## **ESCRIBIR:**

Aquí hay dos formas de escritura, una es escribir solo una variable o valor, y la otra permite escribir una lista de variables o valores que el usuario desee

## **DECLARACIÓN DE VARIABLES:**

Estas pueden declararse de cuatro maneras, las cuales son:

- Tipo variable
- Tipo variable = valor
- Tipo variable, variable, etc.
- Tipo variable, variable, etc. =valor

## **REGLAS DE VARIABLES**

Las variables deben de seguir ciertas reglas, el nombre debe de iniciar con una letra minúscula, no puede iniciar con un número ni mayúscula, no permite las variables del tipo *variable\_* ya que debe de ingresarse algo siempre antes y después del “\_”. Estas pueden contener mayúsculas en ellas

## **REGLAS DE VARIABLES CLASE**

Las variables deben de seguir: El nombre debe de iniciar con una letra mayúscula, no puede iniciar con un número ni minúscula, a diferencia de las variables normales, no se permiten variables que contengan “\_”

## **COMENTARIOS**

Pueden escribirse comentarios en el código, estos comentarios deben de estar delimitados por /\* y \*/ , o empezar por // y terminar en el fin de línea, estos no se toman en cuenta en el analizador sintáctico

## **INSTRUCCIONES :**

Se pueden escribir instrucciones en una misma línea, pero estas deberán de ir separadas por un punto y coma

## **TIPO DE DATOS PARA IDENTIFICADORES:**

- Entero
- Real
- Cadena, estas deben de ir entre comillas, por ejemplo: “cadena”
- Booleano, estos pueden ser Verdadero o Falso

## **TIPO DE DATOS PARA FUNCIONES:**

- Entero
- Real
- Cadena
- Booleano
- Nulo, ese tipo es para cuando la función no retorna ningún valor

## **OPERACIONES ARITMETICAS:**

Se tienen:

- Sumas
- Restas
- Multiplicaciones
- Divisiones
- Logaritmos
- Raíz

## **SIGNOS**

Se permiten asignaciones con valores negativos, ejemplo  $a=-4$

## **PRE Y POST**

- Incremento: Se debe de agregar dos símbolos más después de una variable, para aumentar en uno su magnitud
- Decremento: Se debe de agregar dos símbolos menos después de una variable, para disminuir en uno su magnitud

## **OPERADOR LÓGICO**

- And
- Or

## **SIMBOLOS DE AGRUPACIONES**

Esto permite utilizar paréntesis para agrupar operaciones y para los parámetros

## **EXPRESIONES REGULARES DEL LENGUAJE**

- **PALABRAS RESERVADAS TIPOS DATO Y FUNCIONES**

clase = "Clase"

propiedades = "Propiedades"



métodos = "Metodos"

tipo\_propiedades\_metodos

"Publicas"|"Publicos"|"Privadas"|"Privados"|"Protegidos"|"Protegidas"

=

tipo\_entero = "entero"

tipo\_cadena = "cadena"

tipo\_real = "real"

tipo\_booleano = "booleano"

tipo\_nulo = "nulo"

- **PALABRAS RESERVADAS DE IFS**

if = "si"|"Si"

entonces = "entonces"|"Entonces"

sino = "sino"|"Sino"

- **PALABRAS RESERVADAS LECTURA Y ESCRITURA**

leer = "Leer"|"leer"

escribir = "Escribir"|"escribir"

- **PALABRAS RESERVADAS CICLO, FUNCIONES**

devolver = "devolver"

constructor = "Constructor"|"constructor"

destructor = "Destructor"|"destructor"

principal = "principal"|"Principal"

instanciar = "instanciar"

desde = "desde"|"Desde"

mientras = "mientras"|"Mientras"

incrementar = "incrementar"|"Incrementar"

decrementar = "decrementar"|"Decrementar"

hacer = "hacer"|"Hacer"

extiende = "extiende"|"Extiende"

eliminar = "Eliminar"|"eliminar"

incluir = "incluir"|"Incluir"

nuevo = "nuevo"

- **TIPOS DE DATOS**

numerosDecimales =  $[1-9][0-9]^* | 0$

numerosReales =  $[0-9]^*.[0-9]+[1-9]$

booleano = "verdadero"|"falso"

cadena =  $["\"]([a-zA-Z][0-9][\ ])^*["\"]$

- **REGLAS EXTRAS**

tabulacion =  $["\ "]{4}$

variables =  $[a-z]+\{reglas\_variablesextra\}$

reglas\_variablesextra =  $[[a-z][0-9][A-Z]]^*|[[a-z][0-9][A-Z]]^* \_ "[a-z][0-9][A-Z]]+$

excepciones\_variables =  $[a-z]+[[a-z][0-9][A-Z]]^* \_ " | [A-Z]+\{variables\}$

variables\_clase =  $[A-Z]+\{reglas\_variablesextra\}$

excepciones\_variables\_clase =  $[a-z]+[[a-z][0-9][A-Z]]^* \_ "$

- **SIGNOS**

finInstruccion = ","

parentesis1 = "("

parentesis2 = ")"

coma = ","

suma = "+"

resta = "-"

multiplicacion = "\*\*"

division = "/"

modulo = "%"

exponenciacion = "^"

igual = "="

incremento = "++"

decremento = "--"

and = "AND"

or = "OR"

mayorQue = ">"

menorQue = "<"

igualQue = "=="

distintoQue = "!="

dosPuntos = ":"

corchete1 = "["

corchete2 = "]"

puntoYComa = ";"

punto = "."

- **FUNCIONES ESPECIALES**

cadenaEntero = "cadenaAEntero"

cadenaReal = "cadenaAReal"

cadenaBooleano = "cadenaABoleano"

seno = "seno"

coseno = "coseno"

tangente = "tangente"

logaritmo = "logaritmo"

raiz = "raiz"

- **IGNORAR**

findelinea = \n|\r|\r\n

ignorar = \n|\r\n|\r\n|\" \"

- **REGLAS SINTACTICAS**

Símbolos terminales:

- CLASE
- PROPIEDADES
- METODOS
- IF
- ENTONCES
- SINO
- LEER
- ESCRIBIR
- DEVOLVER
- CONSTRUCTOR

- DESTRUCTOR
- PRINCIPAL
- INSTANCIAR
- DESDE
- MIENTRAS
- INCREMENTO
- DECREMENTO
- HACER
- EXTIENDE
- PARENTESIS1
- PARENTESIS2
- COMA
- IGUAL
- AND
- OR
- DOS\_PUNTOS
- RAIZ
- NUEVO
- MENOS
- PUNTO
- ELIMINAR
- TIPO\_DATO\_CUP
- PALABRA\_RESERVADA\_TIPO
- SUMA
- RESTA
- MULTIPLICACION
- DIVISION
- EXPONENCIACION
- MODULO
- MENOR\_QUE
- MAYOR\_QUE
- DISTINTO\_QUE
- IGUAL\_QUE
- INCREMENTAR
- DECREMENTAR
- SENO
- COSENO
- TANGENTE
- LOGARITMO
- PROPIEDADES\_METODOS
- CADENA\_A\_BOOLEANO
- CADENA\_A\_REAL
- CADENA\_A\_ENTERO
- VARIABLE
- VARIABLE\_CLASE

- Símbolos no terminales

- Nodo DECLARACION\_CLASES\_CUP
- CLASES\_CUP
- ENCABEZADO\_CUP
- PROPIEDADES\_CUP
- METODOS\_CUP
- VARIABLES\_PROPIEDADES\_CUP
- CODIGO\_METODOS\_CUP
- DECLARACION\_VARIABLES\_CUP
- CONDICION\_CUP
- ELSE\_CUP
- VARIAS\_VARIABLES\_CUP
- OPERACIONES\_CUP
- REALIZAR\_OPERACION\_CUP
- LISTA\_OPERACIONES\_CUP
- VARIABLES\_CUP
- VALOR\_INICIAL\_CUP
- ESTRUCTURA\_SELECTIVA\_CUP
- LECTURA\_ESCRITURA\_CUP
- LECTURA\_CUP
- ESCRITURA\_CUP
- DEVOLUCION\_CUP
- OPERADORES\_LOGICOS\_CUP
- TIPOS\_DATOS\_CONDICIONES\_CUP
- DESDE\_CUP
- INC\_DEC\_VAR
- HACER\_CUP
- FUNCIONES\_SUELTAS\_CUP
- LISTA\_VARIABLES\_CUP
- INSTANCIACION\_CUP
- CONSTRUCTOR\_CUP
- DESTRUCTOR\_CUP
- PROGRAMA\_PRINCIPAL\_CUP
- LLAMADA\_METODO\_CLASE\_CUP
- FUNCIONES\_PREDEFINIDAS\_CUP
- CLASE\_GENERAL\_CUP

- Símbolo inicial
  - S

- **DECLARACIÓN DE GRAMÁTICA**

**S** -> CLASE\_GENERAL\_CUP

CLASE\_GENERAL\_CUP -> CLASE\_GENERAL\_CUP CLASES\_CUP  
|CLASES\_CUP

CLASES\_CUP -> ECLARACION\_CLASES\_CUP

DECLARACION\_CLASES\_CUP -> ENCABEZADO\_CUP  
PROPIEDADES\_CUP METODOS\_CUP

ENCABEZADO\_CUP -> CLASE VARIABLE\_CLASE EXTIENDE  
VARIABLE\_CLASE  
|CLASE VARIABLE\_CLASE

PROPIEDADES\_CUP -> PROPIEDADES  
PROPIEDADES\_METODOS DOS\_PUNTOS  
VARIABLES\_PROPIEDADES\_CUP  
PROPIEDADES\_CUP  
|PROPIEDADES PROPIEDADES\_METODOS  
DOS\_PUNTOS VARIABLES\_PROPIEDADES\_CUP

VARIABLES\_PROPIEDADES\_CUP ->  
PALABRA\_RESERVADA\_TIPO  
VARIABLE  
VARIABLES\_PROPIEDADES\_CUP  
  
|VARIABLE\_CLASE VARIABLE  
|PALABRA\_RESERVADA\_TIPO VARIABLE

METODOS\_CUP ->       METODOS PROPIEDADES\_METODOS  
DOS\_PUNTOS CODIGO\_METODOS\_CUP  
METODOS\_CUP  
|METODOS PROPIEDADES\_METODOS  
DOS\_PUNTOS CODIGO\_METODOS\_CUP

CODIGO\_METODOS\_CUP ->   DECLARACION\_VARIABLES\_CUP  
CODIGO\_METODOS\_CUP  
|LECTURA\_ESCRITURA\_CUP  
CODIGO\_METODOS\_CUP  
|ESTRUCTURA\_SELECTIVA\_CUP  
CODIGO\_METODOS\_CUP  
|DESDE\_CUP CODIGO\_METODOS\_CUP  
|HACER\_CUP CODIGO\_METODOS\_CUP  
|FUNCIONES\_SUELTAS\_CUP  
CODIGO\_METODOS\_CUP  
|INSTANCIACION\_CUP  
CODIGO\_METODOS\_CUP  
|CONSTRUCTOR\_CUP  
CODIGO\_METODOS\_CUP  
|DESTRUCTOR\_CUP  
CODIGO\_METODOS\_CUP  
| VARIAS\_VARIABLES\_CUP  
CODIGO\_METODOS\_CUP  
|LLAMADA\_METODO\_CLASE\_CUP  
CODIGO\_METODOS\_CUP  
|LLAMADA\_METODO\_CLASE\_CUP  
|PROGRAMA\_PRINCIPAL\_CUP  
|VARIAS\_VARIABLES\_CUP  
| INSTANCIACION\_CUP  
| FUNCIONES\_SUELTAS\_CUP  
| HACER\_CUP  
| DESDE\_CUP  
| ESTRUCTURA\_SELECTIVA\_CUP

|LECTURA\_ESCRITURA\_CUP  
|REALIZAR\_OPERACION\_CUP  
|DECLARACION\_VARIABLES\_CUP

LLAMADA\_METODO\_CLASE\_CUP -> VARIABLE PUNTO VARIABLE  
PARENTESIS1  
LISTA\_VARIABLES\_CUP  
PARENTESIS2  
|VARIABLE PUNTO VARIABLE  
PARENTESIS1 PARENTESIS2

DECLARACION\_VARIABLES\_CUP -> PALABRA\_RESERVADA\_TIPO  
VARIABLES\_CUP  
VALOR\_INICIAL\_CUP  
  
|PALABRA\_RESERVADA\_TIPO  
VARIABLES\_CUP  
|VARIABLE  
|VARIABLE\_CLASE PUNTO  
VARIABLE VALOR\_INICIAL\_CUP  
|VARIABLE VALOR\_INICIAL\_CUP

VARIABLES\_CUP -> VARIABLE COMA VARIABLES\_CUP  
  
|TIPO\_DATO\_CUP COMA  
VARIABLES\_CUP  
|VARIABLE  
| TIPO\_DATO\_CUP

VALOR\_INICIAL\_CUP -> IGUAL DECREMENTO VARIABLE  
  
|IGUAL INCREMENTO VARIABLE  
  
|IGUAL VARIABLE DECREMENTO  
|IGUAL VARIABLE INCREMENTO  
|IGUAL MENOS VARIABLE  
| IGUAL MENOS TIPO\_DATO\_CUP  
|IGUAL MENOS  
REALIZAR\_OPERACION\_CUP  
|IGUAL VARIABLE  
|IGUAL TIPO\_DATO\_CUP  
|IGUAL REALIZAR\_OPERACION\_CUP  
|IGUAL CONDICION\_CUP  
|IGUAL NUEVO VARIABLE\_CLASE



PARENTESIS1 PARENTESIS2  
|IGUAL FUNCIONES\_PREDEFINIDAS\_CUP

OPERACIONES\_CUP -> SUMA  
|RESTA  
|MULTIPLICACION  
|DIVISION  
|MODULO  
|EXPONENCIACION  
|AND  
|OR

REALIZAR\_OPERACION\_CUP -> VARIABLE LISTA\_OPERACIONES\_CUP  
|TIPO\_DATO\_CUP  
LISTA\_OPERACIONES\_CUP

LISTA\_OPERACIONES\_CUP -> OPERACIONES\_CUP VARIABLE  
LISTA\_OPERACIONES\_CUP  
  
|OPERACIONES\_CUP TIPO\_DATO\_CUP  
LISTA\_OPERACIONES\_CUP  
|OPERACIONES\_CUP TIPO\_DATO\_CUP  
| OPERACIONES\_CUP VARIABLE

LECTURA\_ESCRITURA\_CUP -> LECTURA\_CUP  
|ESCRITURA\_CUP

LECTURA\_CUP -> LEER VARIABLE

ESCRITURA\_CUP -> ESCRIBIR VARIABLES\_CUP

OPERADORES\_LOGICOS\_CUP -> MENOR\_QUE  
|MAYOR\_QUE  
|IGUAL\_QUE  
|DISTINTO\_QUE  
|MODULO

CONDICION\_CUP -> TIPOS\_DATOS\_CONDICIONES\_CUP  
OPERADORES\_LOGICOS\_CUP  
TIPOS\_DATOS\_CONDICIONES\_CUP  
|LLAMADA\_METODO\_CLASE\_CUP

TIPOS\_DATOS\_CONDICIONES\_CUP -> VARIABLE  
|TIPO\_DATO\_CUP

ESTRUCTURA\_SELECTIVA\_CUP -> IF CONDICION\_CUP ENTONCES  
CODIGO\_METODOS\_CUP  
ELSE\_CUP  
|IF CONDICION\_CUP ENTONCES  
CODIGO\_METODOS\_CUP

ELSE\_CUP -> SINO CONDICION\_CUP ENTONCES  
CODIGO\_METODOS\_CUP ELSE\_CUP  
|SINO CONDICION\_CUP ENTONCES  
CODIGO\_METODOS\_CUP  
|SINO CODIGO\_METODOS\_CUP  
CONDICION\_CUP ENTONCES  
CODIGO\_METODOS\_CUP

DEVOLUCION\_CUP -> DEVOLVER TIPO\_DATO\_CUP  
|DEVOLVER VARIABLE

INC\_DEC\_VAR -> INCREMENTAR TIPO\_DATO\_CUP  
|DECREMENTAR TIPO\_DATO\_CUP

DESDE\_CUP -> DESDE DECLARACION\_VARIABLES\_CUP  
MIENTRAS CONDICION\_CUP INC\_DEC\_VAR  
HACER CODIGO\_METODOS\_CUP

HACER\_CUP -> HACER CODIGO\_METODOS\_CUP MIENTRAS  
CONDICION\_CUP

LISTA\_VARIABLES\_CUP -> COMA LISTA\_VARIABLES\_CUP  
|PALABRA\_RESERVADA\_TIPO VARIABLE  
LISTA\_VARIABLES\_CUP  
|PALABRA\_RESERVADA\_TIPO VARIABLE  
|VARIABLE

FUNCIONES\_SUELTAS\_CUP -> PALABRA\_RESERVADA\_TIPO VARIABLE  
PARENTESIS1 LISTA\_VARIABLES\_CUP  
PARENTESIS2 CODIGO\_METODOS\_CUP  
DEVOLUCION\_CUP  
|PALABRA\_RESERVADA\_TIPO VARIABLE  
PARENTESIS1 PARENTESIS2  
CODIGO\_METODOS\_CUP  
DEVOLUCION\_CUP

FUNCIONES\_PREDEFINIDAS\_CUP -> CADENA\_A\_ENTERO PARENTESIS1  
VARIABLE PARENTESIS2  
|CADENA\_A\_REAL PARENTESIS1  
VARIABLE PARENTESIS2  
|CADENA\_A\_BOOLEANO  
PARENTESIS1VARIABLE  
PARENTESIS2  
|RAIZ PARENTESIS1 VARIABLE  
PARENTESIS2  
|TANGENTE PARENTESIS1  
VARIABLE PARENTESIS2  
|COSENO PARENTESIS1 VARIABLE  
PARENTESIS2  
|SENO PARENTESIS1 VARIABLE

PARENTESIS2  
|LOGARITMO PARENTESIS1 VARIABLE  
PARENTESIS2

INSTANCIACION\_CUP -> VARIABLE\_CLASE VARIABLE IGUAL  
INSTANCIAR VARIABLE\_CLASE  
PARENTESIS1 PARENTESIS2  
|VARIABLE IGUAL INSTANCIAR  
VARIABLE\_CLASE PARENTESIS1  
PARENTESIS2

VARIAS\_VARIABLES\_CUP -> ELIMINAR VARIAS\_VARIABLES\_CUP  
  
|ELIMINAR VARIABLE

CONSTRUCTOR\_CUP -> CONSTRUCTOR PARENTESIS1  
LISTA\_VARIABLES\_CUP PARENTESIS2  
CODIGO\_METODOS\_CUP  
|CONSTRUCTOR PARENTESIS1  
PARENTESIS2 CODIGO\_METODOS\_CUP

DESTRUCTOR\_CUP -> DESTRUCTOR PARENTESIS1 PARENTESIS2  
VARIAS\_VARIABLES\_CUP

PROGRAMA\_PRINCIPAL\_CUP -> PALABRA\_RESERVADA\_TIPO  
PRINCIPAL PARENTESIS1  
LISTA\_VARIABLES\_CUP  
PARENTESIS2  
CODIGO\_METODOS\_CUP  
DEVOLUCION\_CUP  
  
| PALABRA\_RESERVADA\_TIPO PRINCIPAL  
PARENTESIS1 PARENTESIS2  
CODIGO\_METODOS\_CUP  
DEVOLUCION\_CUP