

Universidad Rafael Landívar  
Facultad de Ingeniería  
Ingeniería en Informática Y Sistemas  
Compiladores  
Ing. Dhaby Xiloj



## **Informe Proyecto 2da Fase**

Luis José Rodríguez Pérez 15030-18

Kelvin Alejandro Cano Bolaños 16623-12

Quetzaltenango 18 de Noviembre de 2020

## **Informe Sobre El Proyecto**

A continuación, pasaremos a detallar todo lo que utilizamos para realizar el proyecto y así sea más entendible el funcionamiento y la lógica de este:

### **FUNCIONES**

No pertenecen a ninguna clase en específico, sino que son siempre públicas, para declararlas es necesario declarar su tipo, darles un nombre, y si se desea pasarles parámetros. Deben de seguir las reglas de las variables. También pueden retornar un valor que vaya de acuerdo con el tipo de esta y los argumentos son opcionales.

### **FUNCIONES ESPECIALES**

Estas ya están definidas por el mismo lenguaje y están:

- FcI = cadenaAentero

Recibe una cadena y la convierte en un entero.

- FcF = cadenaAreal

Recibe una cadena y la convierte a un número real.

- FcB = cadenaAbooleano

Convierte una cadena a booleano.

### **FUNCIONES MATEMATICAS**

- SEN = seno
- COS = coseno
- TAN = tangente
- LOG = logaritmo
- SQRT = raiz

### **ESTRUCTURAS ITERATIVAS**

Se utilizan para ejecutar instrucciones una cantidad determinada de veces que se le indique, entre ellos podemos mencionar:

- FRD = desde:

Comienza con la palabra “desde”, luego puede estar una variable o una inicialización de una variable a un valor, luego debe de ir la palabra “mientras”, luego la condición; cabe aclarar que 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 realizaran mientras se cumpla la condición

- FRH hacer:

Comienza con la palabra “hacer”, luego deben de ir las instrucciones que se desean, la palabra “mientras” y una condición.

## **CLASES**

En este lenguaje nos encontraremos 2 formas de declarar una clase, la primera 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 reconoce como una clase, si no como un identificador. Cada clase puede tener sus propiedades y métodos, las propiedades deberán de ir declaradas de primero y luego los métodos. También pueden incluir lo que son los constructores y destructores.

### **• Propiedades**

Son 3 distintas las cuales son:

1. CLPRPU = propiedades publicas.
2. CLPRPV = propiedades privadas.
3. CLPRPT = propiedades protegidas.

### **• Métodos**

Son 3 distintos los cuales son:

1. CLMPU = metodos publicos.
2. CLMPV = metodos privados.
3. CLMPT = metodos protegidos.

- MetRes = constructor.
- MetRes = destructor.

## **INSTANCIACION (CLIN)**

Se utiliza par instanciar clases ya declaradas palabra reservada “instanciar” (CLIN),

## **HERENCIA (CLEX)**

Se utilizan para extender funciones de una clase por medio de la palabra reservada “instanciar” (CLEX).

## **FUNCION PRINCIPAL (METPRIN)**

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 retornar algún valor.

### **Condiciones**

- MYQ = Mayor que
- MNQ = Menor que
- EQ = Igual
- NEQ = Distinto que

## **IDENTACION (IDEN)**

Identar correctamente el código es esencial para el buen funcionamiento del programa.

## **LEER (RD)**

Solo puede leerse una variable por vez.

## **ESCRIBIR (WT)**

El lenguaje solo tiene dos formas para escribir, escribir solo una variable o valor, y la otra permite escribir un listado de variables o valores que el usuario desee.

## **DECLARACIÓN DE VARIABLES**

Se pueden declarar de las siguientes 4 maneras:

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

## **REGLAS DE VARIABLES**

Para que la correcta sintaxis de las mismas el nombre debe de iniciar con una letra minúscula, no puede iniciar con un número ni mayúscula, pero si permite las mayúsculas en medio de ellas ejemplo vaRiable.

## **REGLAS DE VARIABLES CLASE**

Para una correcta sintaxis el nombre debe de iniciar con una letra mayúscula, no puede iniciar con un número ni minúscula.

## **COMENTARIOS**

Para que los comentarios no produzcan errores en el código deben estar delimitados por /\* y \*/ , o empezar por // y terminar en el fin de línea.

## **TIPO DE DATOS PARA IDENTIFICADORES**

- INT = entero
- FLOAT = real
- STRING = cadena NOTA: estas obligatoriamente deben ir entre comillas: "texto"
- BOOL = booleano

## **TIPO DE DATOS PARA FUNCIONES**

- INT = entero
- FLOAT = real
- STRING = cadena
- BOOL = booleano

## **SIGNOS**

- INC = incremento (++)
- DEC = decremento (--)

## **SIMBOLOS DE AGRUPACIONES**

PIZ = (

PDR = )

## LEXEMAS UTILIZADOS

METPRIN	= "Principal"
INT	= "entero"
STRING	= "cadena"
BOOL	= "booleano"
FLOAT	= "real"
IMP	= "incluir"
IF	= "si"
THN	= "entonces"
ELS	= "sino"
RETN	= "devolver"
CL	= "clase"
CLEX	= "extiende"
CLPRPU	= "propiedades publicas"
CLPRPT	= "propiedades protegidas"
CLPRPV	= "propiedades privadas"
CLMTPU	= "metodos publicos"
CLMTPT	= "metodos protegidos"
CLMTPV	= "metodos privados"
CLIN	= "instanciar"
FRD	= "desde"
FRM	= "mientras"
FRH	= "hacer"
FRINC	= "incrementar"
FRDEC	= "decrementar"
RD	= "leer"
WT	= "escribir"

SPC	= [ \t\f]
F	= \r \n \r\n
P	= "."
COMA	= ","
INCR	= "++"
DECR	= "--"
SUM	= "+"
RES	= "-"
MUL	= "*"
DIV	= "/"
MOD	= "%"
POT	= "^"
EQ	= "="
EQEQ	= "=="
NEQ	= "!="
MYQ	= ">"
MNQ	= "<"
PIZ	= "("
PDR	= ")"
DSP	= ":"
FLC	= ";"
OLOG	= (AND OR NOT)
MetRes	= (constructor destructor)
NBOOL	= "TRUE" "FALSE"
NUM	= 0 (-?)[1-9][0-9]*
NMCL	= [A-Z][a-zA-Z0-9_]+
NUMR	= (-?)[0-9]+{P}[0-9]+
IDEN	= {let}({NUM}?{let}?)*

CADTXT = {com}({let}|{SPC}|{letdig}|{lqs2})\*{com}  
 FcF = {FLOAT}{SPC}"cadenaAreal"  
 FcI = {INT}{SPC}"cadenaAentero"  
 FcB = {BOOL}{SPC}"cadenaAbooleano"  
 SEN = {FLOAT}{SPC}"seno"  
 COS = {FLOAT}{SPC}"coseno"  
 TAN = {FLOAT}{SPC}"tangente"  
 LOG = {FLOAT}{SPC}"logaritmo"  
 SQRT = {FLOAT}{SPC}"raiz"

/\*ERRORES TOKENS\*/

Noldes = ({lqs2}){NUM}){IDEN} | {IDEN}{lqs2}

La definición de la gramática, con sus cuatro elementos principales **G = {T, N, S, P}**.

#### **T (Símbolos terminales):**

terminal IF, THN, ELS, INT, STRING, BOOL, FLOAT, FRD, FRM, FRH, FRINC, FRDEC, NBOOL, NUM, NUMR, IDEN, CADTXT, NMCL, CL, CLEX, CLPRPU, CLPRPT, CLPRPV, CLMTPU, CLMTPT, CLMTPV, MetRes, METPRIN, IMP, FcF, FcI, FcB, SEN, COS, TAN, LOG, SQRT, RETN, CLIN, RD, WT, SPC, P, COMA, INCR, DECR, POT, EQ, EQEQ, NEQ, MYQ, MNQ, PIZ, PDR, FLC, OLOG, SUM, RES, MUL, DIV, MOD, DSP;

#### **N (Símbolos no terminales):**

LOOP, LineaC, Import\_lib, Gramar\_clase, FuncPrin, Decl\_clase, Prop\_clase, Metod\_clase, STRUC\_if, ifU, ifELS, cond1, cond2, cond3, cond4, cond5, DeclPrin, CodSuelto, Pinea, Retor, Salida, Asign, Declar, Metodo, ESTRUCTURA\_INSTANCIAR, OPERACIONES\_CICLOS, SIGNOS\_COMPARACION, OPERACIONES\_ARITMETICAS, ESTRUCTURA\_FOR, OPERACIONES\_DO, ESTRUCTURA\_DO, ESPACIOS\_5

#### **S = PROGRAMA**

LOOP ::= LineaC LOOP | LineaC



## **P = PRODUCCIONES**

LOOP ::= LineaC LOOP | LineaC;

LineaC ::= Import\_lib | Gramar\_clase FuncPrin | FuncPrin;

Import\_lib ::= IMP SPC CADTXT;

FuncPrin ::= DeclPrin CodSuilt;

DeclPrin ::= INT SPC METPRIN PIZ PDR;

Gramar\_clase ::= Decl\_clase Gramar\_clase | Decl\_clase;

Decl\_clase ::= CL SPC NMCL Prop\_clase | CL SPC NMCL Metod\_clase;

Prop\_clase ::= SPC SPC CLPRPU DSP SPC SPC CodSuilt FLC Prop\_clase |  
SPC SPC CLPRPU DSP  
                  | SPC SPC CLPRPV DSP SPC SPC CodSuilt FLC Prop\_clase | SPC SPC  
CLPRPV DSP  
                  | SPC SPC CLPRPT DSP SPC SPC CodSuilt FLC Prop\_clase | SPC SPC  
CLPRPT DSP;

Metod\_clase ::= SPC SPC CLMTPU DSP SPC SPC CodSuilt FLC Metod\_clase |  
SPC SPC CLMTPU DSP  
                  | SPC SPC CLMTPV DSP SPC SPC CodSuilt FLC Metod\_clase | SPC  
SPC CLMTPV DSP  
                  | SPC SPC CLMTPT DSP SPC SPC CodSuilt FLC Metod\_clase | SPC  
SPC CLMTPT DSP;

CodSuilt ::= Pinea CodSuilt | Pinea;

Pinea ::= SPC SPC Salida

- | SPC SPC RD SPC IDEN
- | SPC SPC Retor
- | SPC SPC Asign
- | SPC SPC Declar
- | SPC SPC Metodo
- | SPC SPC STRUC\_if
- | SPC SPC ESTRUCTURA\_FOR
- | SPC SPC ESTRUCTURA\_DO
- | SPC SPC ESTRUCTURA\_INSTANCIAR;

Retor ::= RETN SPC NUM;

Salida ::= WT SPC CADTXT

- | WT SPC IDEN
- | WT SPC CADTXT COMA SPC IDEN
- | WT SPC IDEN COMA SPC CADTXT
- | WT SPC IDEN COMA SPC CADTXT COMA SPC IDEN
- | WT SPC IDEN COMA SPC NUM COMA SPC CADTXT
- | WT SPC Metodo
- | WT SPC Metodo COMA SPC Metodo
- | WT SPC Metodo COMA SPC IDEN;

Asign ::= IDEN SPC EQ SPC NUM

- | IDEN SPC EQ SPC CADTXT
- | IDEN SPC EQ SPC NUMR

| IDEN SPC EQ SPC IDEN OPERACIONES\_ARITMETICAS NUM  
 | IDEN SPC EQ SPC NUM OPERACIONES\_ARITMETICAS IDEN  
 | IDEN SPC EQ SPC NUM OPERACIONES\_ARITMETICAS NUM  
 | IDEN SPC EQ SPC IDEN  
 | Metodo EQ SPC NUM;

Metodo ::= IDEN P IDEN PIZ IDEN PDR | IDEN P IDEN PIZ PDR | IDEN P IDEN;

OPERACIONES\_ARITMETICAS ::= SUM

|RES

|MUL

|MOD

|DIV;

OPERACIONES\_CICLOS ::= INCR

|DECR

|FRDEC

|FRINC;

SIGNOS\_COMPARACION ::= EQEQ

|NEQ

|MYQ

|MNQ;

ESTRUCTURA\_FOR ::= FRD SPC IDEN SPC FRM SPC IDEN SPC  
 SIGNOS\_COMPARACION SPC NUM SPC OPERACIONES\_CICLOS SPC NUM  
 SPC FRH SPC SPC SPC SPC WT SPC IDEN

|FRD SPC IDEN SPC EQ SPC NUM SPC FRM SPC IDEN SPC  
 SIGNOS\_COMPARACION SPC NUM SPC FRH SPC SPC SPC SPC WT

SPC IDEN SPC SPC SPC SPC SPC IDEN SPC EQ SPC IDEN  
OPERACIONES\_ARITMETICAS NUM

|FRD SPC IDEN SPC EQ SPC NUM SPC FRM SPC IDEN SPC  
SIGNOS\_COMPARACION SPC NUM SPC FRH SPC SPC SPC SPC WT  
SPC IDEN SPC SPC SPC SPC SPC IDEN SPC EQ SPC IDEN  
OPERACIONES\_CICLOS;

OPERACIONES\_DO ::= IDEN EQ NUM OPERACIONES\_ARITMETICAS NUM  
OPERACIONES\_DO

|IDEN EQ IDEN OPERACIONES\_ARITMETICAS IDEN  
OPERACIONES\_DO

|IDEN EQ IDEN OPERACIONES\_ARITMETICAS NUM  
OPERACIONES\_DO

|IDEN EQ NUM OPERACIONES\_ARITMETICAS IDEN  
OPERACIONES\_DO

|IDEN EQ NUM OPERACIONES\_ARITMETICAS NUM

|IDEN EQ IDEN OPERACIONES\_ARITMETICAS IDEN

|IDEN EQ IDEN OPERACIONES\_ARITMETICAS NUM

|IDEN EQ NUM OPERACIONES\_ARITMETICAS IDEN;

ESTRUCTURA\_DO ::= FRH SPC SPC SPC SPC SPC OPERACIONES\_DO SPC  
SPC FRM SPC IDEN SPC SIGNOS\_COMPARACION SPC IDEN

|FRH SPC SPC SPC SPC SPC OPERACIONES\_DO SPC SPC FRM  
SPC IDEN SPC SIGNOS\_COMPARACION SPC NUM

|SPC SPC FRH SPC SPC SPC SPC SPC OPERACIONES\_DO SPC  
SPC FRM SPC IDEN SPC SIGNOS\_COMPARACION SPC IDEN

|SPC SPC FRH SPC SPC SPC SPC SPC OPERACIONES\_DO SPC  
SPC FRM SPC IDEN SPC SIGNOS\_COMPARACION SPC NUM;

ESTRUCTURA\_INSTANCIAR ::= NMCL SPC IDEN SPC EQ SPC CLIN SPC  
NMCL PIZ PDR

|IDEN SPC EQ SPC CLIN SPC NMCL PIZ PDR;

Declar ::= INT SPC Assign | FLOAT SPC Assign | STRING SPC Assign | INT SPC  
IDEN SPC | STRING SPC IDEN SPC | NMCL SPC IDEN SPC ;

STRUC\_if ::= ifU STRUC\_if | ifELS STRUC\_if |ifU | ifELS ;

ifU ::= IF SPC cond1 SPC THN SPC SPC CodSueIt FLC SPC SPC SPC SPC  
RETN SPC NUM

    | IF SPC cond2 SPC THN SPC SPC CodSueIt FLC SPC SPC SPC SPC  
RETN SPC NUM

    | IF SPC cond3 SPC THN SPC SPC CodSueIt FLC SPC SPC SPC SPC  
RETN SPC NUM

    | IF SPC cond4 SPC THN SPC SPC CodSueIt FLC SPC SPC SPC SPC  
RETN SPC NUM

    | IF SPC cond5 SPC THN SPC SPC CodSueIt FLC SPC SPC SPC SPC  
RETN SPC NUM;

ifELS ::= ELS SPC cond1 SPC THN SPC SPC SPC SPC RETN SPC NUM

    | ELS SPC cond2 SPC THN SPC SPC SPC SPC RETN SPC NUM

    | ELS SPC cond3 SPC THN SPC SPC SPC SPC RETN SPC NUM

    | ELS SPC cond4 SPC THN SPC SPC SPC SPC RETN SPC NUM

    | ELS SPC cond5 SPC THN SPC SPC SPC SPC RETN SPC NUM

    | ELS SPC SPC SPC SPC RETN SPC NUM;

cond1 ::= IDEN MOD NUM;

cond2 ::= IDEN EQEQ NUM;

cond3 ::= IDEN NEQ NUM;

cond4 ::= IDEN MYQ NUM;

cond5 ::= IDEN MNQ NUM;

