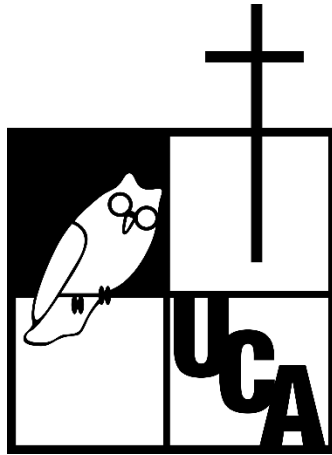


Universidad Centroamericana José Simeón Cañas



Definición de Lenguaje: CPrime

Materia

Compiladores

Catedrático

Lic. Néstor Aldana

Estudiantes

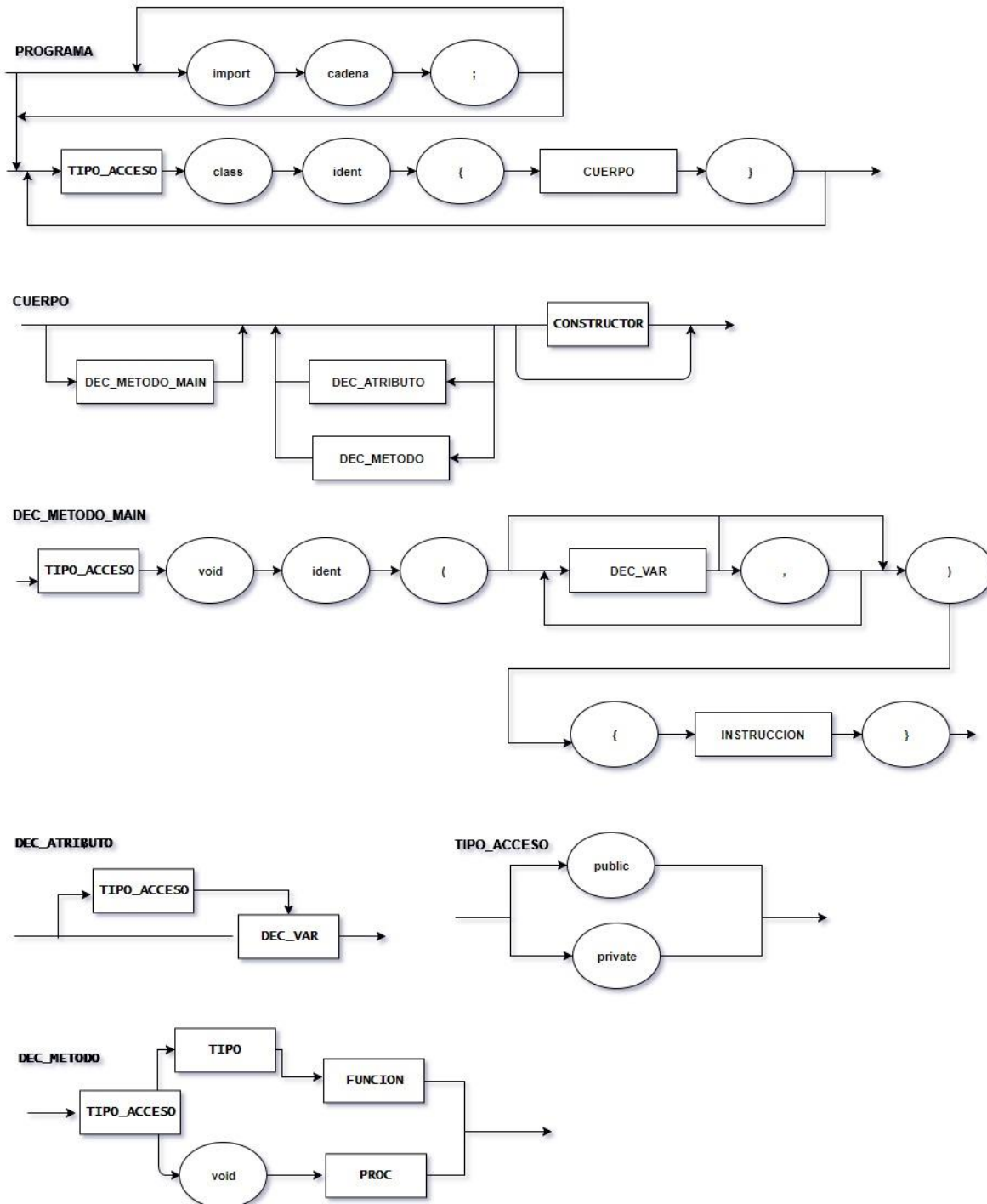
Méndez Hernández Melvin Edgardo	00064713
Ramos Hernández Marlon David	00004912
Polanco Zelada José Orlando	00052813

Antiguo Cuscatlán 31 de enero de 2018

Definición del Leguaje de programación: CPrime

La razón por el cual se eligió el nombre es porque la letra “C” representa uno de los leguajes de programación en el cual se han derivado otros lenguajes hoy en día son de alto nivel y que permiten desarrollar grandes aplicaciones que son utilizadas en el ámbito empresarial en el cual permite simular una situación de la vida cotidiana que se puede resolver a partir de las propiedades y comportamiento de un tipo de objeto concreto, además porque fue el primer lenguaje de programación que se utilizó en la carrera universitaria. Prime en su traducción al español significa ‘Principal’, por lo tanto, CPrime se da a entender que el lenguaje de programación ‘C’ es el núcleo principal para desarrollar el lenguaje del curso de compiladores.

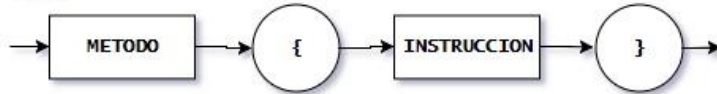
Diagrama de sintaxis



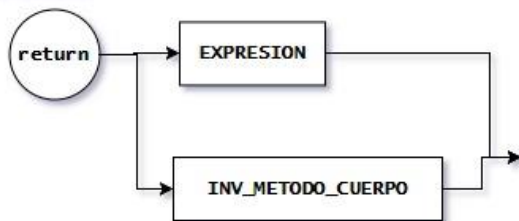
FUNCION



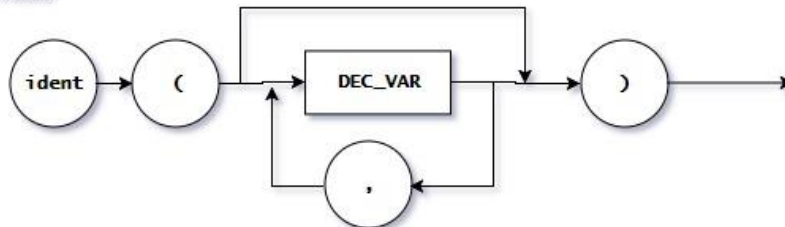
PROC



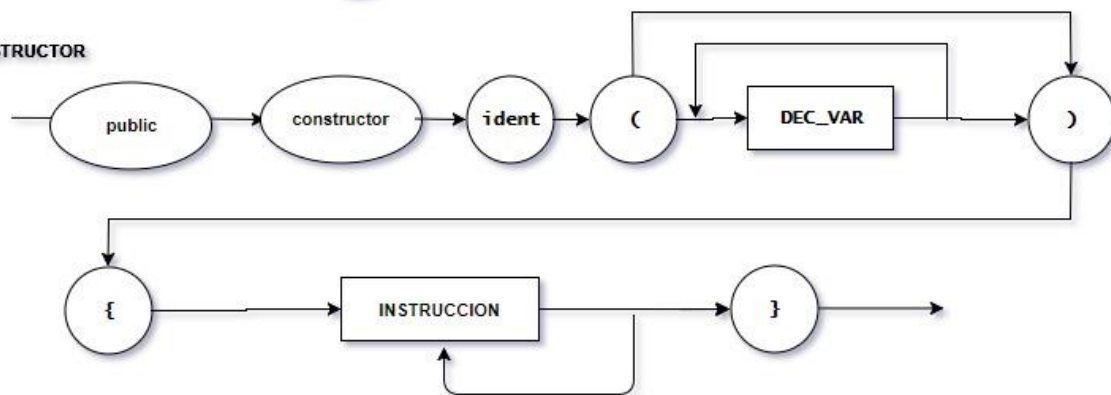
RETORNAR



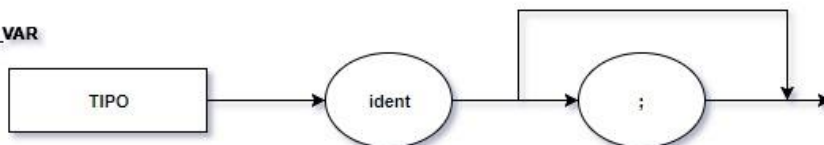
METODO

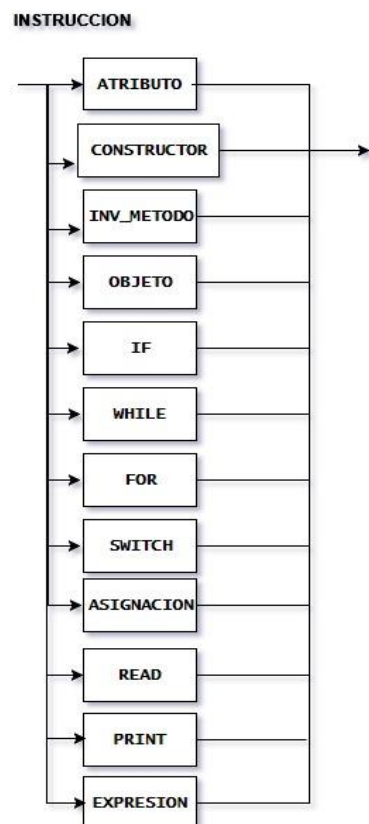
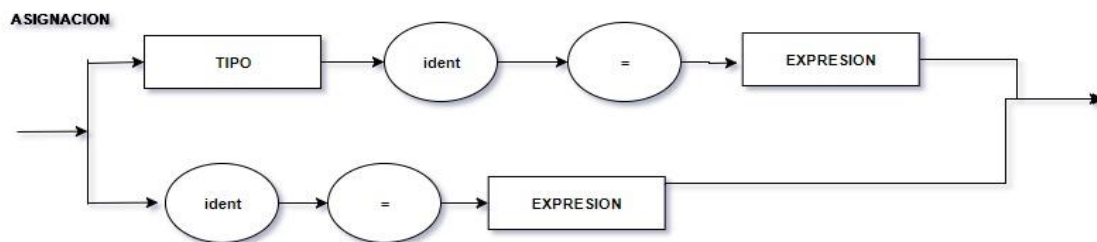
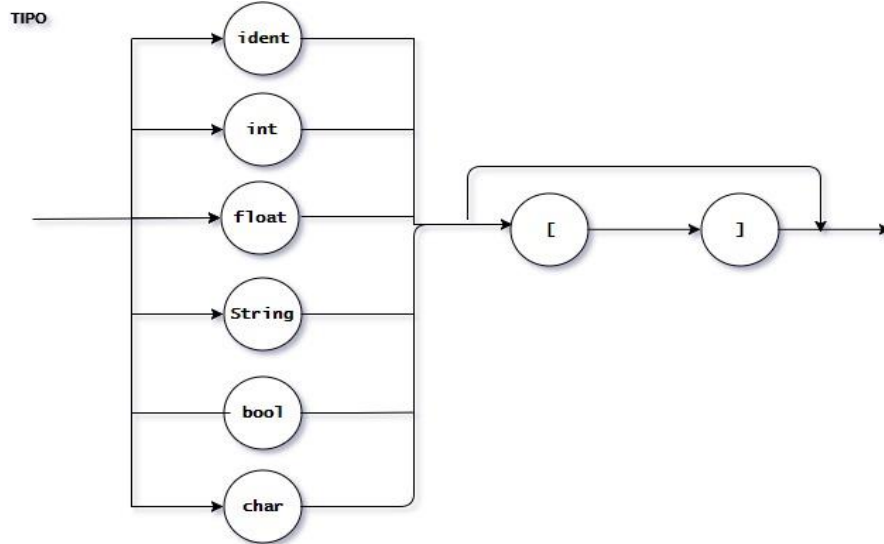


CONSTRUCTOR

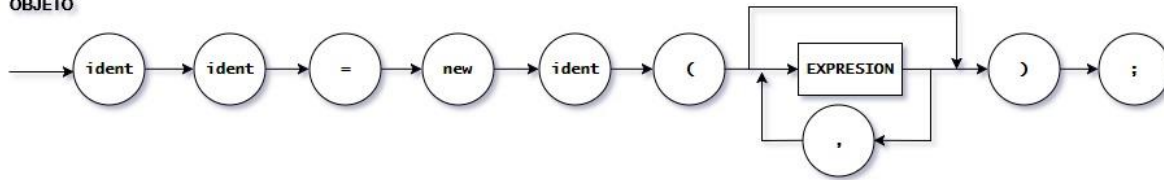


DEC_VAR

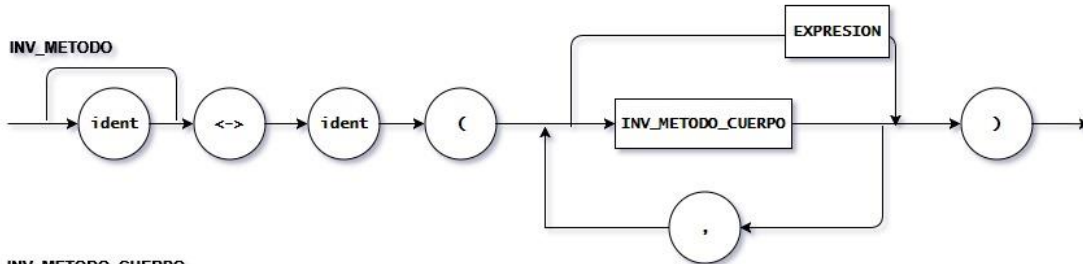




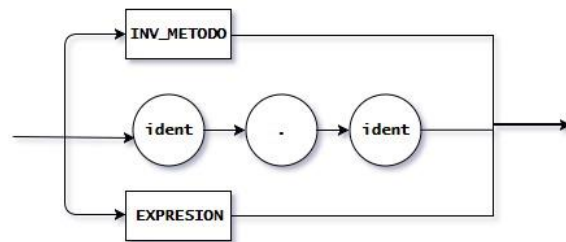
OBJETO

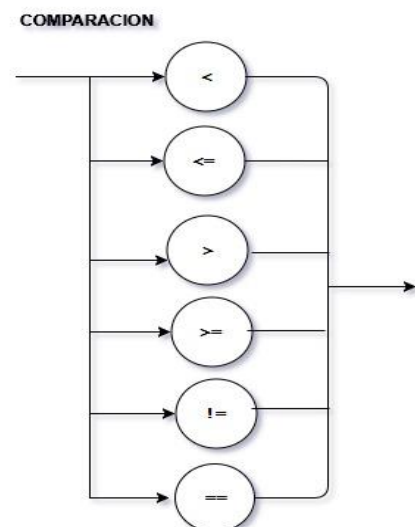
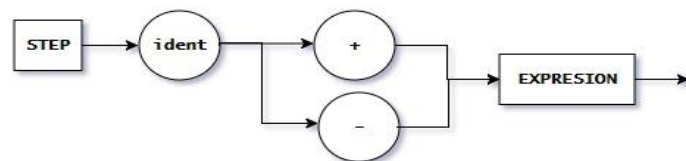
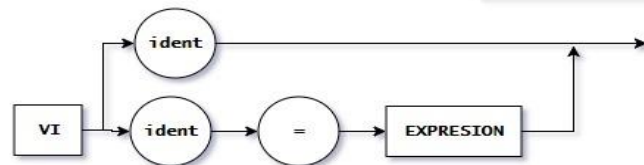
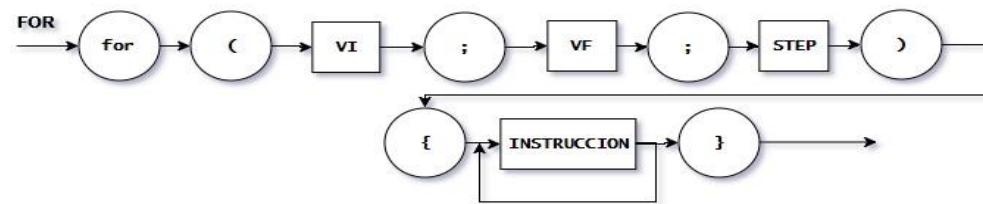
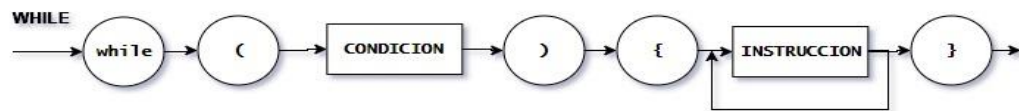
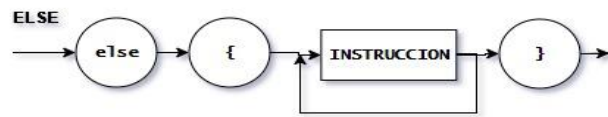
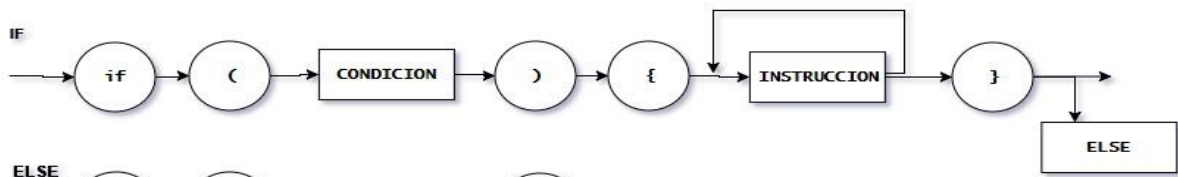


INV_METODO

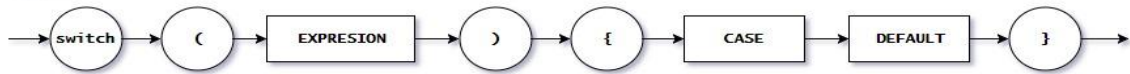


INV_METODO_CUERPO

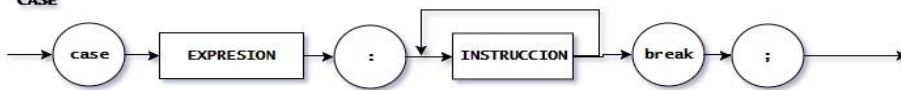




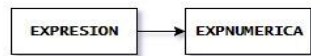
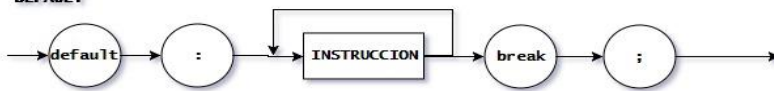
SWITCH



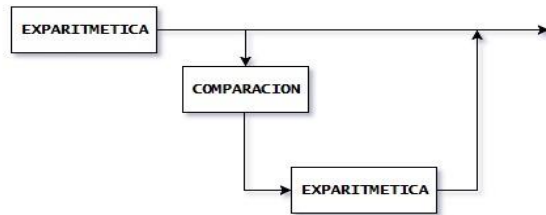
CASE



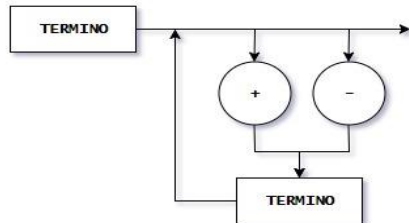
DEFAULT



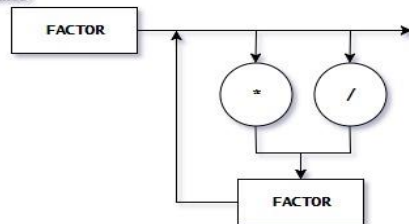
EXPNUMERICA



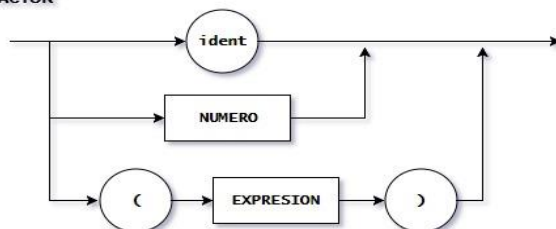
EXPARITMETICA

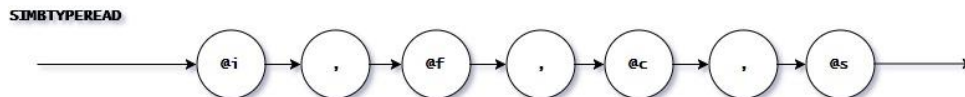
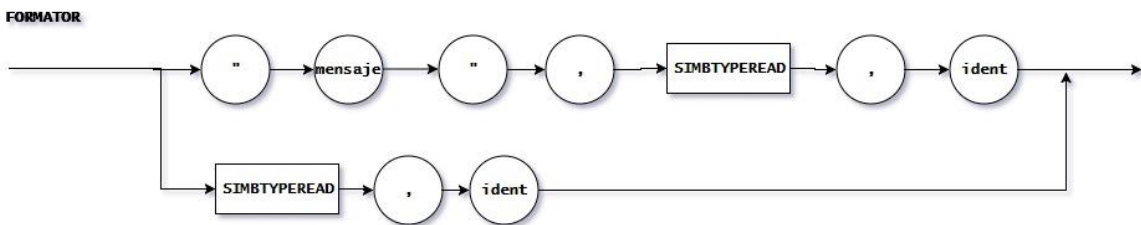
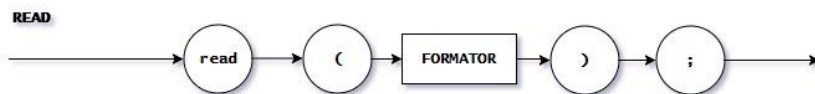
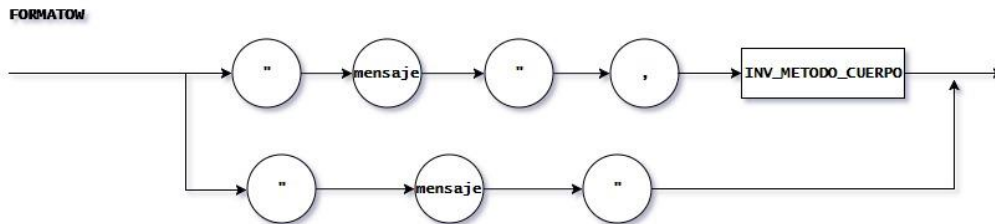
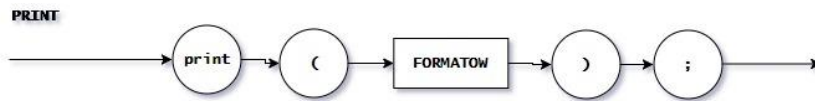
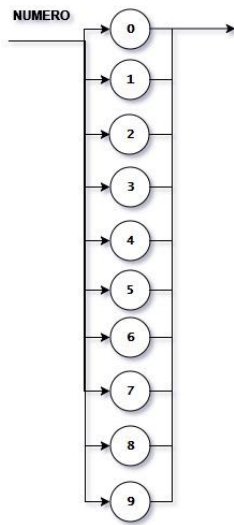


TERMINO



FACTOR





Gramática del lenguaje

#	SIMBOLO NO TERMINAL	PRODUCCIONES
1	PROGRAMA	Import cadena;* TIPO_ACCESO class ident { CUERPO }+
2	CUERPO	DEC_METODO_MAIN? DEC_ATRIBUTO* DEC_METODO* CONSTRUCTOR? e
3	DEC_METODO_MAIN	TIPO_ACCESO void ident '\'(DEC_VAR DEC_VAR, e)'\'* { INSTRUCCION }
4	DEC_ATRIBUTO	TIPO_ACCESO? DEC_VAR
5	DEC_METODO	TIPO_ACCESO TIPO FUNCION TIPO_ACCESO void PROC
6	TIPO_ACCESO	public private
7	FUNCION	METODO { INSTRUCCION RETORNAR }
8	RETORNAR	return EXPRESION return INV_METODO_CUERPO
9	PROC	METODO { INSTRUCCIÓN }
10	METODO	ident '\'(DEC_VAR, DEC_VAR e)'\'*
11	CONSTRUCTOR	public constructor ident (DEC_VAR e)* { INSTRUCCIÓN+ }
12	DEC_VAR	TIPO ident; TIPO ident
13	TIPO	ident int float String bool char ident[] int[] float[] String[] bool[] char[]
14	ASIGNACION	TIPO ident = EXPRESION ident = EXPRESION
15	INSTRUCCION	ATRIBUTO CONSTRUCTOR INV_METODO OBJETO IF WHILE FOR SWITCH ASIGNACION READ PRINT EXPRESION
16	OBJETO	ident ident = new ident'\'(EXPRESION, EXPRESION e)'\'*;
17	INV_METODO	ident? <-> ident '\'(EXPRESION INV_METODO_CUERPO, INV_METODO_CUERPO)'\'+
18	INV_METODO_CUERPO	INV_METODO ident.ident EXPRESION
19	IF	if (CONDICION) { INSTRUCCION+ } ELSE*
20	ELSE	else {INSTRUCCION+}
21	WHILE	while (CONDICION) { INSTRUCCION+ }
22	FOR	for (VI ; VF ; STEP) { INSTRUCCION+ }
23	VI	ident = EXPRESION ident
24	VF	ident COMPARACIO EXPRESION
25	STEP	ident + EXPRESION ident - EXPRESION
26	COMPARACION	< <= > >= != ==
27	SWITCH	switch '\'(EXPRESION)'\' { CASE+ DEFAULT}
28	CASE	case EXPRESION: INSTRUCCION+ break;
29	DEFAULT	default : INSTRUCCION+ break;
30	CONDICION	EXPRESION COMPARACION EXPRESION
31	EXPRESION	EXPNUMERICA
32	EXPNUMERICA	EXPARITMETICA EXPARITMETICA COMPARACION EXPARITMETICA
33	EXPARITMETICA	TERMINO TERMINO + TERMINO TERMINO - TERMINO
34	TERMINO	FACTOR FACTOR * FACTOR FACTOR / FACTOR
35	FACTOR	ident NUMERO '\'(EXPRESION)'\'
36	NUMERO	0 1 2 3 4 5 6 7 8 9
37	PRINT	print(FORMATOW);

38	FORMATOW	"cadena", INV_METODO_CUERPO "cadena"
39	READ	read '\'(FORMATOR)'\';
40	FORMATOR	"mensaje", SIMBTYPEREAD , ident SIMBTYPEREAD , ident //ident es la variable para almacenar
41	SIMBTYPEREAD	@i,@f,@c,@s //i entero, f float, c caracter, s String para identificar que tipo de valor ha leído

Palabras reservadas del lenguaje

private	int
void	float
public	String
class	char
if	bool
else	step
switch	new
case	print
default	read
while	
for	

Tabla de expresiones regulares

Simbología:

() : paréntesis agrupación

[] : corchetes significa el operador de ó

'-' : guion indica un rango

Expresión regular	Descripción
# [a-zA-Z0-9] ? #	Esta expresión regular indica los comentarios de bloque que se definirá en el lenguaje
# [a-zA-Z0-9] ?	Indica que solo será un comentario de línea

Códigos ejemplos según nuestro lenguaje

Código 1: cálculo de factorial

```
1 start factorialN{
2   public class main {
3     int n;
4     int i;
5     int factorial;
6     print("Ingrese el numero para sacar su factorial");
7     read (@i, n);
7     for (i=1 ; i<=n ; i++) {
8       factorial = factorial * i;
9     }
10    print("El factorial del numero es:" ,factorial);
11 }
```



```
public void resta(int n1,int n2){  
    return n1 - n2;  
}  
public void suma(int n1,int n2){  
    return n1 * n2;  
}  
public void multiplicacion(int n1,int n2){  
    return n1 * n2;  
}  
public void division(int n1,int n2){  
    return n1 / n2;  
}  
  
}  
end
```

Código 3: obtener y mostrar la información de Estudiantes

```
start inicClase {  
    public class main(){  
        int c;  
        string nomE;  
        float prom;  
        int ins;  
        read("Ingrese el carne del estudiante:" @i,c);  
        read("Ingrese el nombre del estudiante:" @i,nomE);  
        read("Ingrese el promedio del estudiante:" @i,prom);  
        read("Estudiante inscrito? 1 = si, 0 = no:" @i,ins);  
        estudiante student = estudiante.constructor(c,nomE,prom,ins);  
        print("El carne del estudiante es" student<->carnet );  
        print("El nombre del estudiante es" student<->nomEstudiante);  
        print("El promedio del estudiante es" student<->inscrito);  
        if(exc estudiante<->estainscrito){  
            print("El estudiante esta inscrito");  
        }  
        else{  
            print("El estudiante no esta inscrito");  
        }  
    }  
} end
```

```
public class estudiante{  
    public int carnet;  
    public string nomEstudiante;  
    public float promedio;  
    public int inscrito;  
  
    public constructor estudiante(carnet, nomEstudiante,  
promedio,inscrito){  
        carnet = carnet;  
        nomEstudiante = nomEstudiante;  
        promedio = promedio;  
        inscrito = inscrito;  
    }  
  
    public bool estaInscrito(){  
        if(inscrito == 1) {  
            return true;  
        }else {  
            return false;  
        }  
    }  
}
```


Llamar los métodos de una clase

Para llamar los métodos que están definidos en una clase se utilizara la siguiente sintaxis:

Ident<->nombredemetodo(PARAMETROS)

Ident : es el nombre del objeto instanciado

PARAMETROS: son todos aquellos valores que puede recibir un método, ver la definición de la gramática.

Semántica de objetos

A continuación, se describen algunas funciones para los objetos STRING, INT, FLOAT en el cual se explicará cómo sería la semántica y sintaxis.

TIPO	INSTRUCCIÓN	DESCRIPCION
STRING	VSTRING <-> STRLNG (CADENA)	Para llamar al objeto se escribirá en mayúsculas el nombre del objeto y con los símbolos '<->' se podrá acceder a los métodos como es el caso de poder calcular la longitud de la cadena que se le pase como parámetro, este método solo podrá recibir estrictamente una cadena.
	VSTRING <-> SEARCHC ("CADENA",'CARÁCTER')	Este método recibe como parámetros una cadena y un carácter el cual retornará la posición o índice de ese carácter dentro de la cadena.
	VSTRING <-> REPLACE("CADENA",'OLDCAR','NEWCAR')	Este método realiza un reemplazo de un carácter en una cadena que se le pasen como parámetros. Este método es sensitivo, es decir, si se desea

		reemplazar la letra 'a' (minúscula) reemplaza solo aquellos caracteres que sean 'a' minúscula por la nueva letra.
VINT	VINT<->PARSEF(NUMFLOAT)	Este método solo quita sin aproximación la parte decimal de un numero tipo flotante.
	VINT<->PARIMPAR(NUM)	Solamente recibe como parámetro un número y retorna true o false si el numero es par entonces true caso contrario false.
VFLOAT	VFLOAT<->CUTDECIMAL(NUMFLOTANTE,NUMDIGITOS DESPUESDELPUNTO)	Este método solo expresa un numero flotante en la cantidad de decimales que se desea sin realizar ningún tipo de aproximación.
	VFLOAT<->APROXCUT(NUMFLOTANTE,NUMDIGITOS DESPUESDELPUNTO)	Este método solo expresa un numero flotante en la cantidad de decimales que se desea REALIZANDO aproximación.
	VFLOAT<->PARTDEC(NUMFLOTANTE,Ndecimales)	Este método solo quita la parte entera del numero y solo devuelve los decimales del numero que se pasa como parámetro. El segundo parámetro se especifica el numero de decimales que desea dejar del número, para dejar todos los decimales se escribirá 0.

Librerías

La librería que se utilizará es la matemática será “math.lib” el cual algunas de sus funciones serán.

INSTRUCCIÓN	DESCRIPCION
POTENCIA (BASE,EXPONENTE)	Realiza el calculo para elevar un numero entero a una potencia entera
SENO(NUM),COSENO(NUM),TANGENTE(NUM)	Funciones trigonométricas el parámetro que se le manda es un numero y el resultado será en radianes.
PI	Es un valor constante por lo tanto siempre tendrá predeterminado 3.14159265
FACTORIAL(NUM)	Realiza el calculo de factorial dado un numero entero como parámetro
RAIZCUAD(NUM)	Retorna el cálculo de la raíz cuadrada de un numero que se pase como parámetro.
COMBINACION(N,R)	Calcula la combinación de dos números estrictamente enteros
PERMUTACION(N,R)	Calcula la permutación de dos números estrictamente enteros.
SUMASERIES([1,2,3,4,5])	Realiza la suma de todos los números de una serie.
RESIDUODIV(DIVIDENDO,DIVISOR)	Retorna el residuo de una división en el cual se pasan como parámetros un DIVIDENDO y un DIVISOR que son los elementos de una división
VABSOLUTO(NUM)	Devuelve el valor absoluto de un numero entero o flotante.