A01657055 A00821805

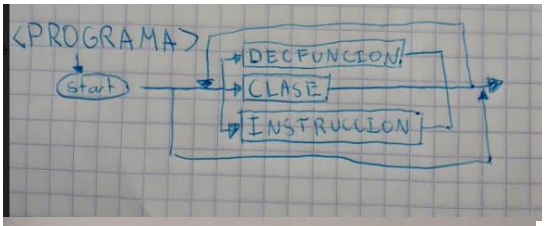
Sofia Recinos Ulrik Alberto Nuño Tapia

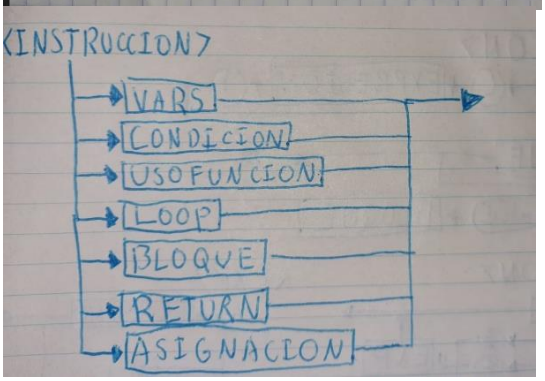
TAREA 3.1

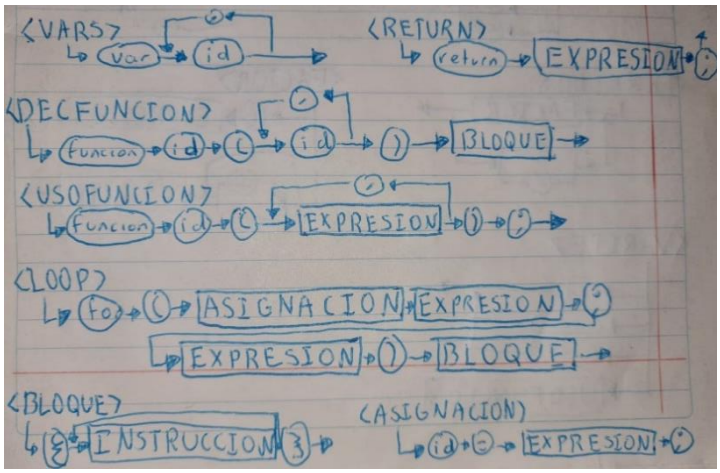
Expresiones regulares:

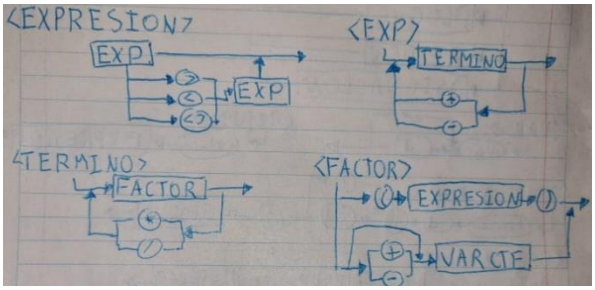
Token	Pattern
start	“start”
id	([A-Z]|[a-z])([A-Z]|[a-z]|\-|_|[0-9])*
;	“;”
{	“{“
}	“}”
,	“ ,”
.	“ .”
:	“:”
<	“<”
>	“>”
class	“class”
print	“print”
cte.string	\” ([^\”] | \\\”)* \”
(	“(“
)	“)”
+	“+”
-	“-“
*****	“*”
/	“/”
var	“var”
for	“for”
function	“function”
cte.i	[0-9]+
cte.f	[0-9]+ \.? [0-9]*

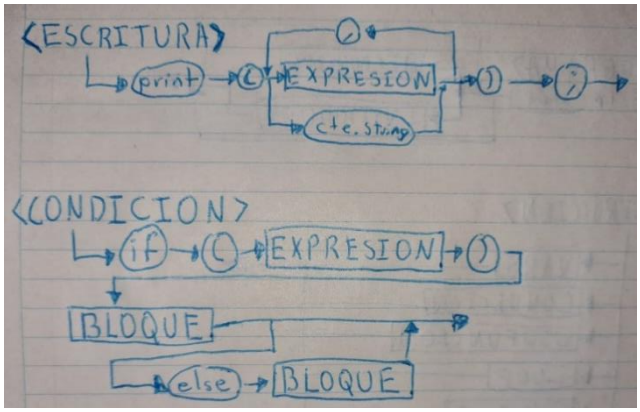
Diagramas:

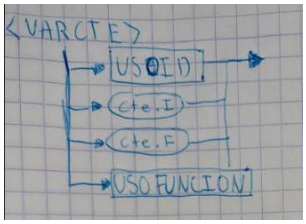

<PROGRAMA>
start
DECFUNCION
CLASE
INSTRUCCION

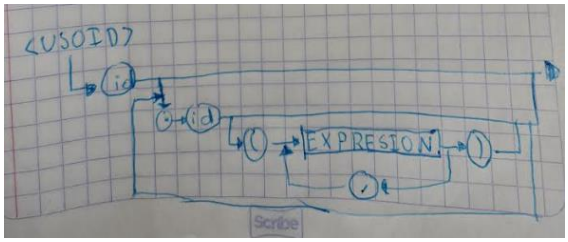

<INSTRUCCION>
VARS
CONDICION
USOFUNCION
LOOP
BLOQUE
RETURN
ASIGNACION


<VARS>
var
id
<RETURN>
return
EXPRESION
<DECFUNCION>
FUNCION
id
(
id
)
BLOQUE
<USOFUNCION>
FUNCION
id
(
EXPRESION
)
;
<LOOP>
for
(
ASIGNACION
EXPRESION
;
EXPRESION
)
BLOQUE
<BLOQUE>
{
INSTRUCCION
}
<ASIGNACION>
id
=
EXPRESION
;


<EXPRESION>
EXP
<
>
=
EXP
<EXP>
TERMINO
+
-
<TERMINO>
FACTOR
*
/
<FACTOR>
(
EXPRESION
)
+
-
VARCTE


<ESCRITURA>
print
(
EXPRESION
cte.String
)
;
<CONDICION>
if
(
EXPRESION
)
BLOQUE
else
BLOQUE


<VARCTE>
USOID
cte.I
cte.F
USOFUNCION


<USOID>
id
.
id
(
EXPRESION
)
,
Scribe

Gramática:

PROGRAMA-> start PROGVARS

PROGVARS-> PROGVAR PROGVARS | ε

PROGVAR-> DECFUNCION | INSTRUCCION | CLASE

INSTRUCCION-> VARS | CONDICION | USOFUNCION | LOOP | BLOQUE | RETURN | ASIGNACION

CLASE-> class id BLOQUEC

BLOQUEC -> { CBLOQUEC }

CBLOQUEC -> CBLOQUEC2 CBLOQUEC | ε

CBLOQUEC2 -> DECFUNCION | VARS

VARS-> var id IDS ;

IDS-> , id IDS| ε

RETURN-> return EXPRESION ;

DECFUNCION-> function id (FVARS) BLOQUE

USOFUNCION-> function id (FVARSU) ;

FVARS -> id IDS| ε

FVARSU -> EXPRESION FVARSUS | ε

FVARSUS -> , EXPRESION FVARSUS | ε

LOOP-> for (ASIGNACION EXPRESION ; EXPRESION) BLOQUE

BLOQUE-> { INSTRUCCIONES }

INSTRUCCIONES-> INSTRUCCION INSTRUCCIONES | ε

ASIGNACION-> id = EXPRESION ;

ESCRITURA-> print (P_ESCRITURA M_ESCRITURA) ;

P_ESCRITURA-> EXPRESION | cte.string

M_SCRITURA -> , P_ESCRITURA M_ESCRITURA | ε

EXPRESION-> EXP POSEXP

POSEXP-> SYMEXP EXP | ε

SYMEXP-> > | < MAY

MAY-> > | ε

EXP-> TERMINO MEXP

MEXP-> SUMRES EXP | ε

SUMRES-> + | -

TERMINO-> FACTOR MTERMINO

MTERMINO-> MULTDIV TERMINO | ε

MULTDIV-> * | /

FACTOR-> (EXPRESION) | SUMRESVAC VAR_CTE

SUMRESVAC-> SUMRES | ε

CONDICION-> if (EXPRESION) BLOQUE ELSE

ELSE-> else BLOQUE | ε

VAR_CTE-> USOID | cte.i | cte.f | USOFUNCION

USOID -> id USOID2

USOID2 -> . id USOID3 USOID2| ε

USOID3 -> (FVARSU) | ε