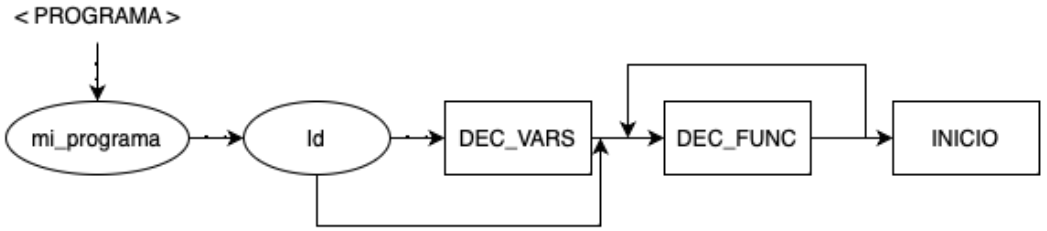
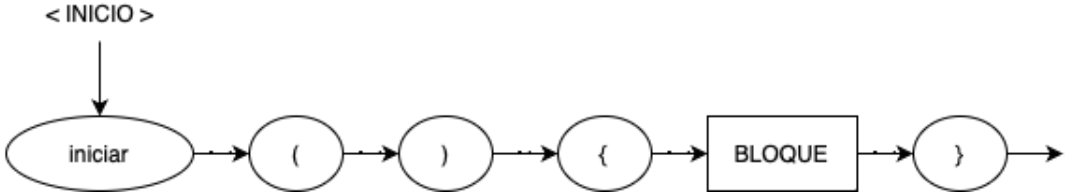
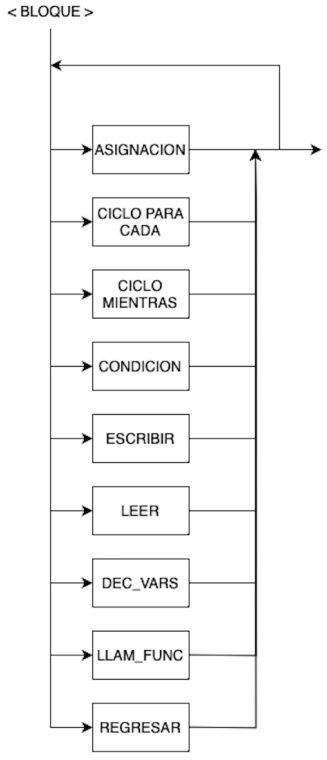
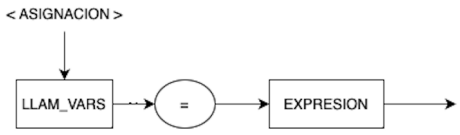
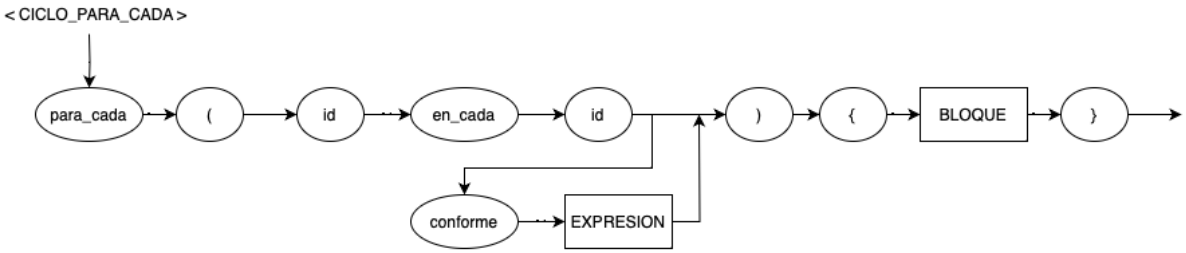
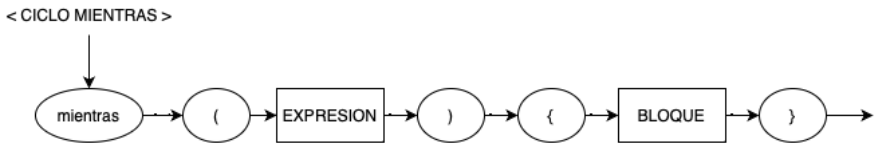

< PROGRAMA >
mi_programa
Id
DEC_VARS
DEC_FUNC
INICIO

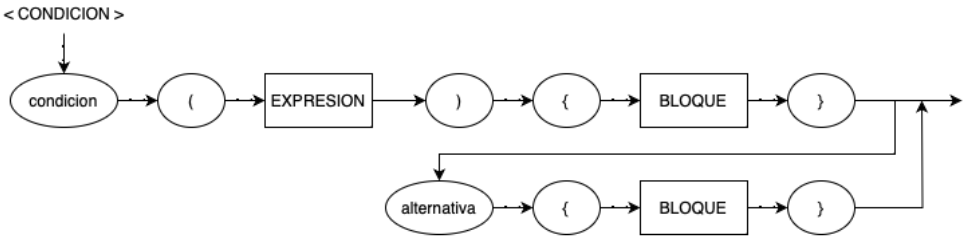

< INICIO >
iniciar
(
)
{
BLOQUE
}

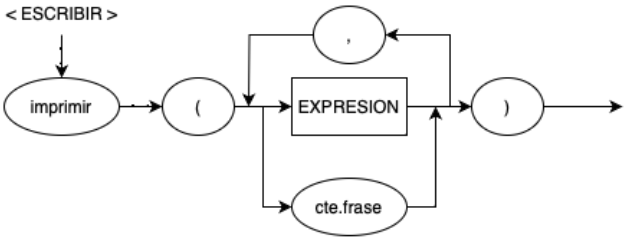

< BLOQUE >
ASIGNACION
CICLO PARA CADA
CICLO MIENTRAS
CONDICION
ESCRIBIR
LEER
DEC_VARS
LLAM_FUNC
REGRESAR

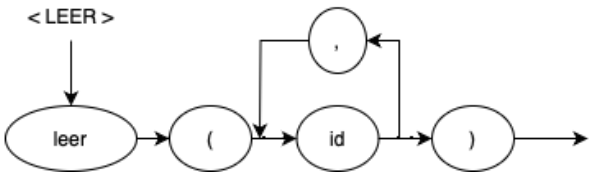

< ASIGNACION >
LLAM_VARS
=
EXPRESION

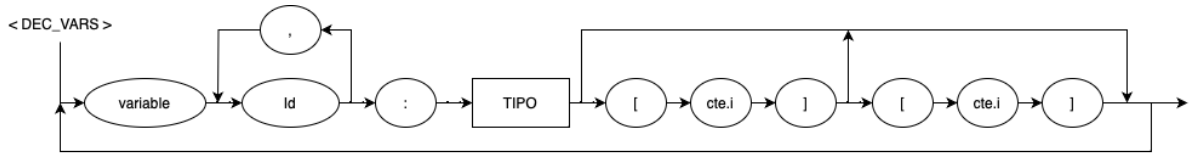

< CICLO_PARA_CADA >
para_cada
(
id
en_cada
id
)
{
BLOQUE
}
conforme
EXPRESION

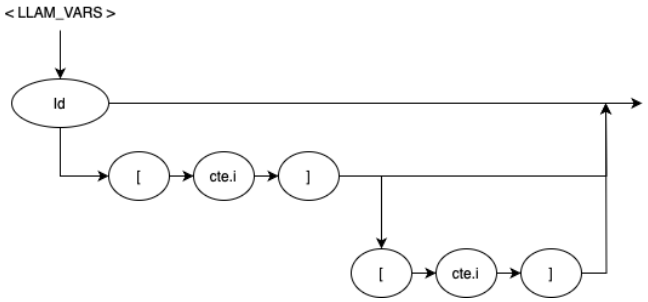

< CICLO MIENTRAS >
mientras
(
EXPRESION
)
{
BLOQUE
}

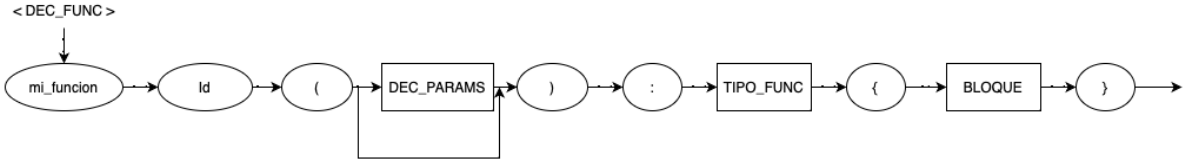

< CONDICION >
condicion
(
EXPRESION
)
{
BLOQUE
}
alternativa
{
BLOQUE
}

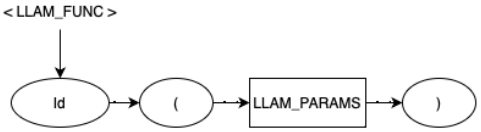

< ESCRIBIR >
imprimir
(
,
EXPRESION
)
cte.frase

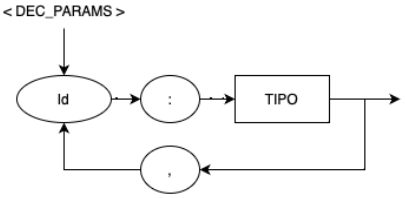

< LEER >
leer
(
,
id
)

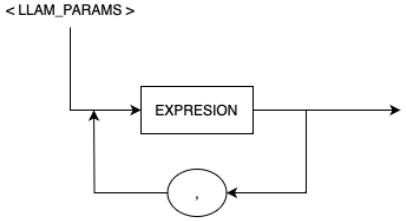

< DEC_VARS >
,
variable
Id
:
TIPO
[
cte.i
]
[
cte.i
]


< LLAM_VARS >
Id
[
cte.i
]
[
cte.i
]

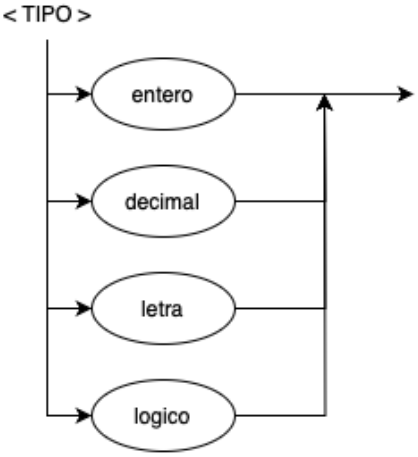

< DEC_FUNC >
mi_funcion
Id
(
DEC_PARAMS
)
:
TIPO_FUNC
{
BLOQUE
}

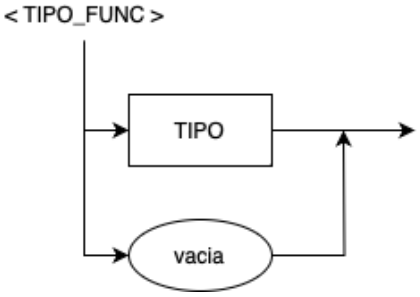

< LLAM_FUNC >
Id
(
LLAM_PARAMS
)

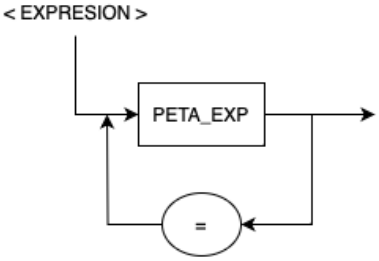

< DEC_PARAMS >
Id
:
TIPO
,

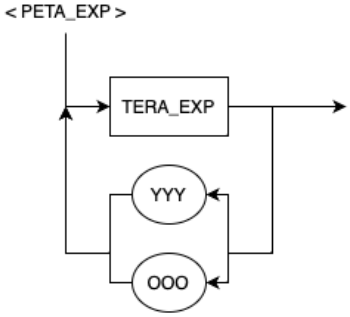

< LLAM_PARAMS >
EXPRESION
,

< REGRESAR >
regresar
(
EXPRESION
)

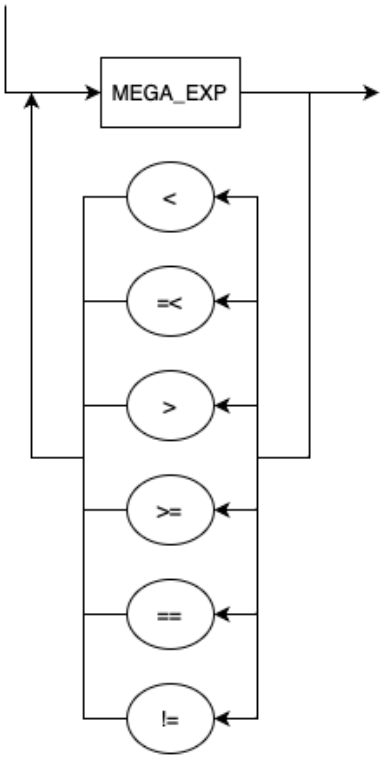

< TIPO >
entero
decimal
letra
logico


< TIPO_FUNC >
TIPO
vacia

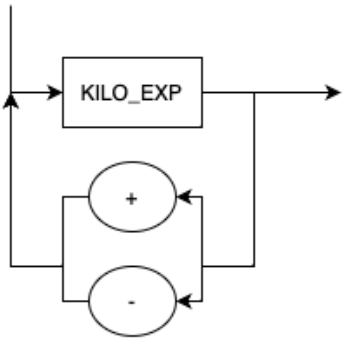

< EXPRESION >
PETA_EXP
=


< PETA_EXP >
TERA_EXP
YYY
OOO

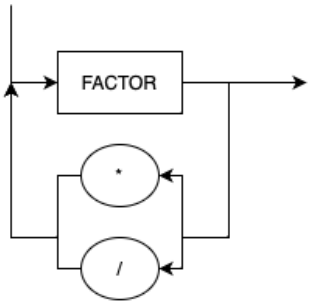
< TERA_EXP >

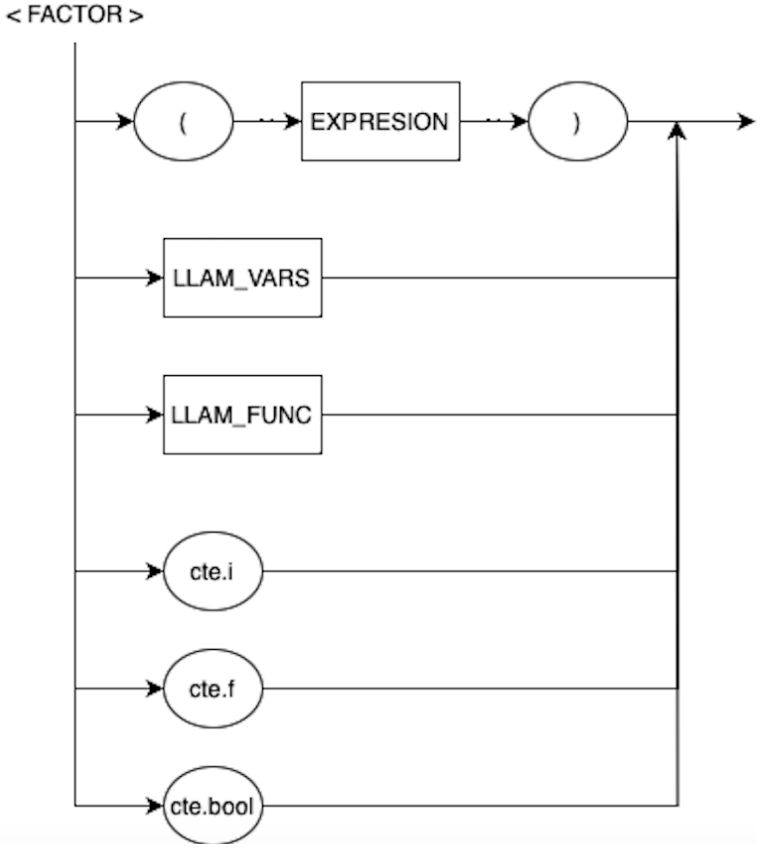


< MEGA_EXP >



< KILO_EXP >




< FACTOR >
(
EXPRESION
)
LLAM_VARS
LLAM_FUNC
cte.i
cte.f
cte.bool