

Gabriel Ribeiro Bernardi – 11821BCC036

Marília Soares Rodrigues – 11821BCC020

Pedro Henrique Gonçalves Teixeira – 11821BCC008

1ª Etapa do Projeto

Especificação da linguagem:

Definição da gramática livre de contexto (GLC) com as estruturas da linguagem especificada

Gramática livre de contexto (GLC)

$S = \langle S \rangle$

$T = \{ \text{program}, (,), \text{begin}, \text{end}, :, ;, \text{if}, \text{then}, \text{else}, \{, \}, \text{while}, \text{repeat}, \text{until}, \text{id}, :=, =, !=, <, >, <=, >=, +, -, *, /, ^, \text{int}, \text{char}, \text{float}, \text{constchar}, \text{constint}, \text{constfloat}, \text{relop} \}$

$N = \{ \langle S \rangle, \langle \text{Bloco} \rangle, \langle \text{DeclaracaoVariavel} \rangle, \langle \text{SequenciaComandos} \rangle, \langle \text{Lista_ids} \rangle, \langle \text{ComandoAtribuicao} \rangle, \langle \text{ComandoRepeticao} \rangle, \langle \text{ComandoCondicional} \rangle, \langle \text{Tipo} \rangle, \langle \text{Condição} \rangle, \langle \text{Expressao} \rangle, \langle \text{CondOP} \rangle \}$

P :

$S \rightarrow \text{program id}() \text{Bloco}$

$\text{sequenciaComandos} \rightarrow \text{ComandoAtribuicao} \mid \text{ComandoCondicional} \mid \text{ComandoRepeticao}$

$\text{Bloco} \rightarrow \text{begin DeclaracaoVariavel SequenciaComandos end}$

$\text{DeclaracaoVariavel} \rightarrow \epsilon \mid \text{tipo} : \text{Lista_ids}$

$\text{Lista_ids} \rightarrow \text{id}; \mid \text{id}, \text{Lista_ids}$

$\text{Tipo} \rightarrow \text{char} \mid \text{float} \mid \text{int}$

$\text{ComandoCondicional} \rightarrow \text{if Condição then Bloco} \mid \text{if Condição then Bloco else Bloco} \mid \text{if Condição then SequenciaComandos} \mid \text{if Condição then Bloco else SequenciaComandos}$

$\text{ComandoRepeticao} \rightarrow \text{while Condição Bloco} \mid \text{repeat Bloco until Condição} ; \mid \text{while Condição sequenciaComandos} \mid \text{repeat SequenciaComandos until Condição} ;$

$\text{ComandoAtribuicao} \rightarrow \text{id} := \text{expressao} ; \mid \text{id} := \text{constchar} ;$

$\text{Condição} \rightarrow \text{Expressao relop Expressao}$

Expressao -> Expressao + Expressao | Expressao - Expressao | Expressao *
Expressao | Expressao / Expressao | Expressao ^ Expressao | id | constint | constfloat |
(Expressao)

Identificação dos tokens usados na gramática

Identificação dos tokens usados na gramática	
Token	Atributo
program	-
id	Posição na tabela de símbolos
(-
)	-
begin	-
end	-
:	-
;	-
if	-
then	-
else	-
{	-
}	-
while	-
repeat	-
until	-
:=	-
=	-
!=	-
<	-
>	-
<=	-
>=	-
+	-
-	-
*	-
/	-
^	-
int	-
char	-
float	-

constint	Valor da constante
constfloat	ponto fixo (PF) ou notação científica (NC)
constchar	Valor da constante
relop	Operador (EQ,NE,LT,LE,GT,GE)

Definição dos padrões (expressões regulares) de cada token (inclusive os tokens especiais)

Definição dos padrões (Expressões Regulares)

Token	Atributo
program	→ program
id	→ [a-zA-Z][a-zA-Z0-9]*
(→ (
)	→)
begin	→ begin
end	→ end
:	→ :
;	→ ;
if	→ if
then	→ then
else	→ else
{	→ {
}	→ }
while	→ while
repeat	→ repeat
until	→ until
:=	→ :=
=	→ =
!=	→ !=
<	→ <
>	→ >
<=	→ <=
>=	→ >=
+	→ +
-	→ -
*	→ *
/	→ /
^	→ ^
int	→ int
char	→ char
float	→ float → float

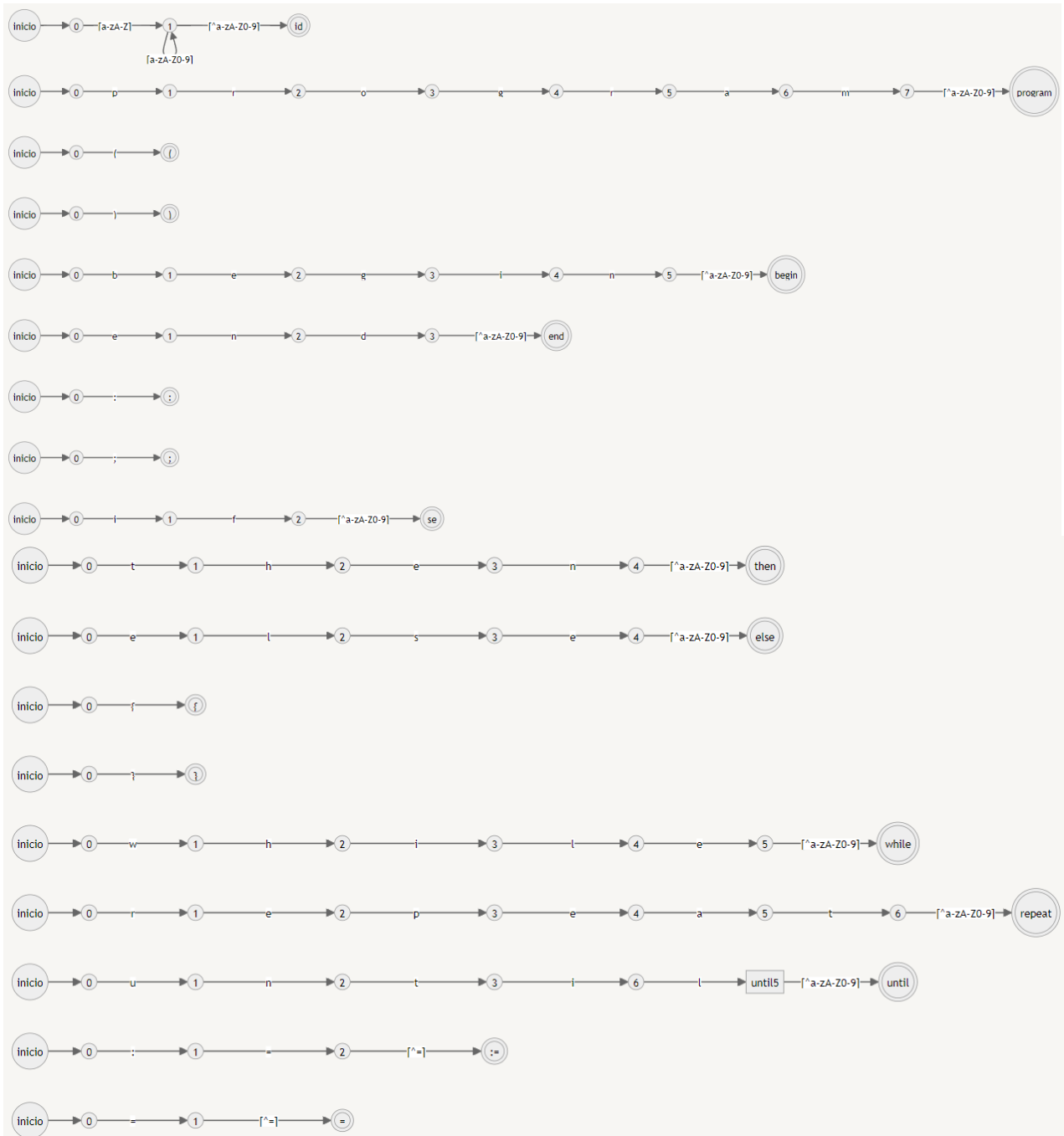
constint	→ [0-9][0-9]*
constfloat	→ [0-9][0-9]*\.[0-9][0-9]*(E[+-]?[0-9][0-9]*)?
constchar	→ '[.]'
relop	→ [= != < > >= <=]

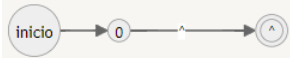
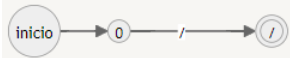
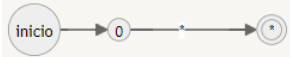
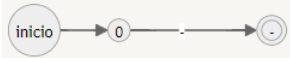
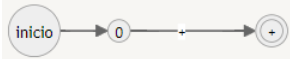
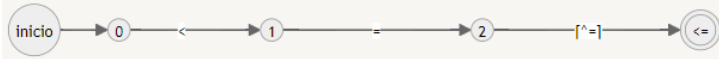
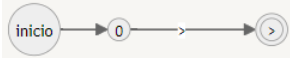
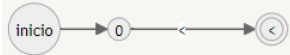
2ª Etapa do Projeto

Análise Léxica (especificação)

Elaboração do diagrama de transição

Gerar um diagrama de transição para cada token





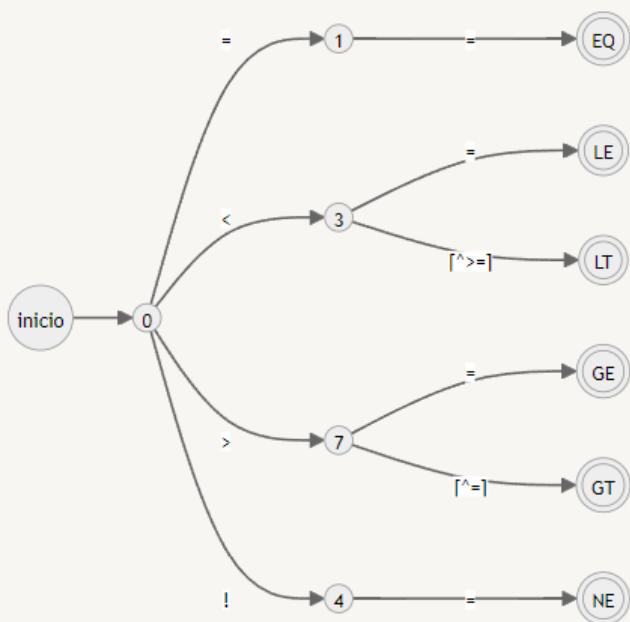
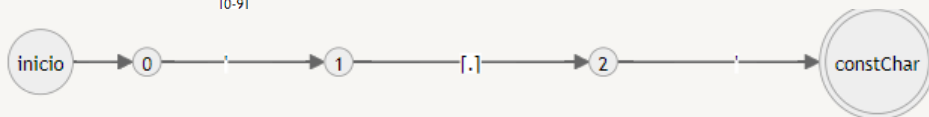
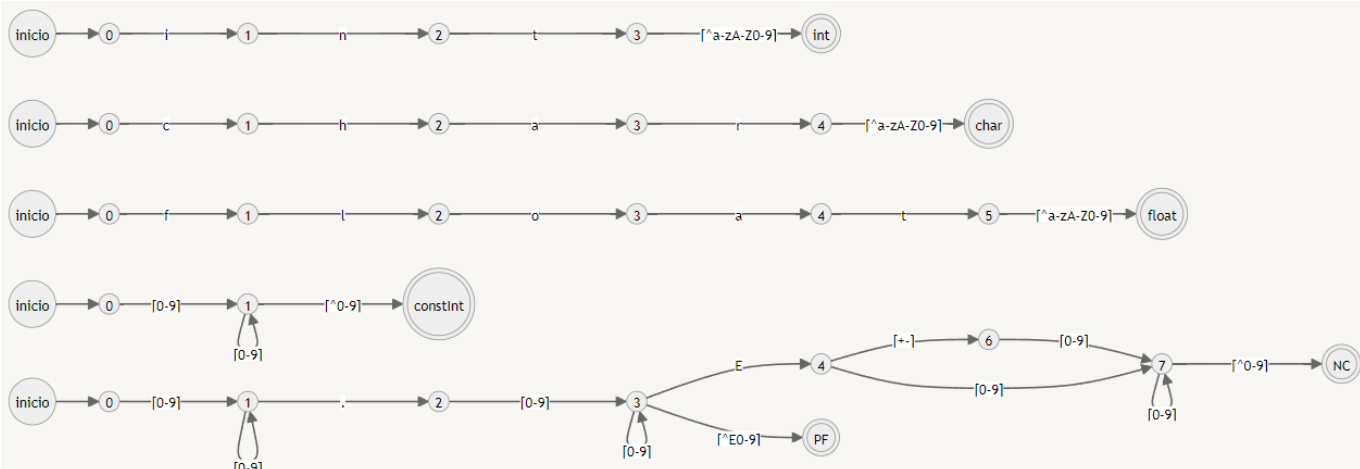


Diagrama não determinístico:

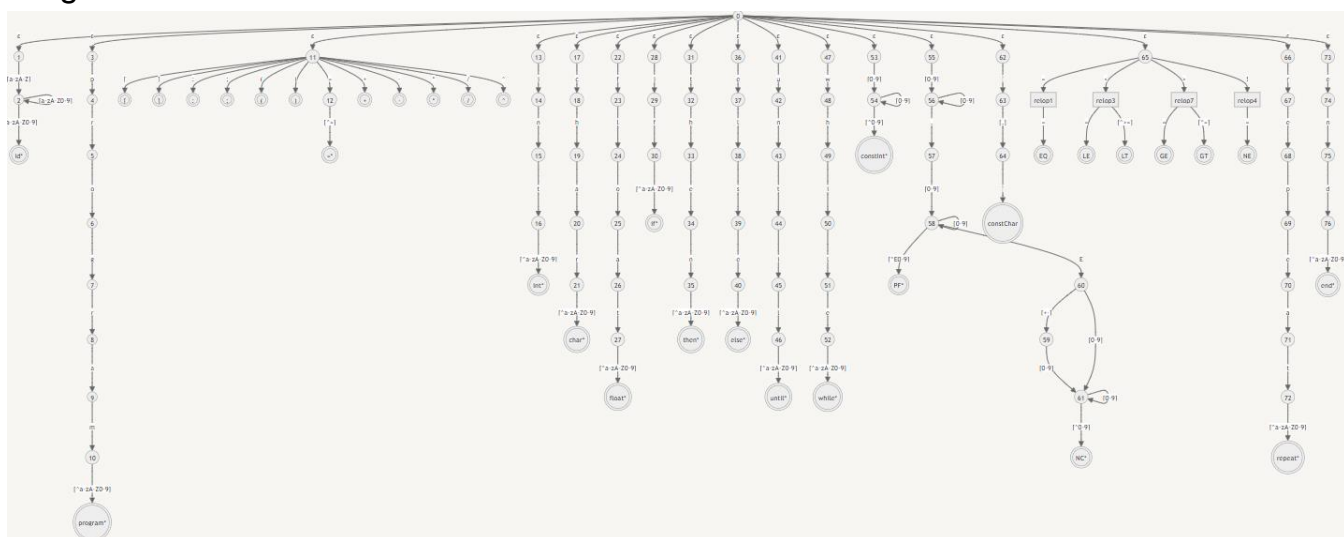
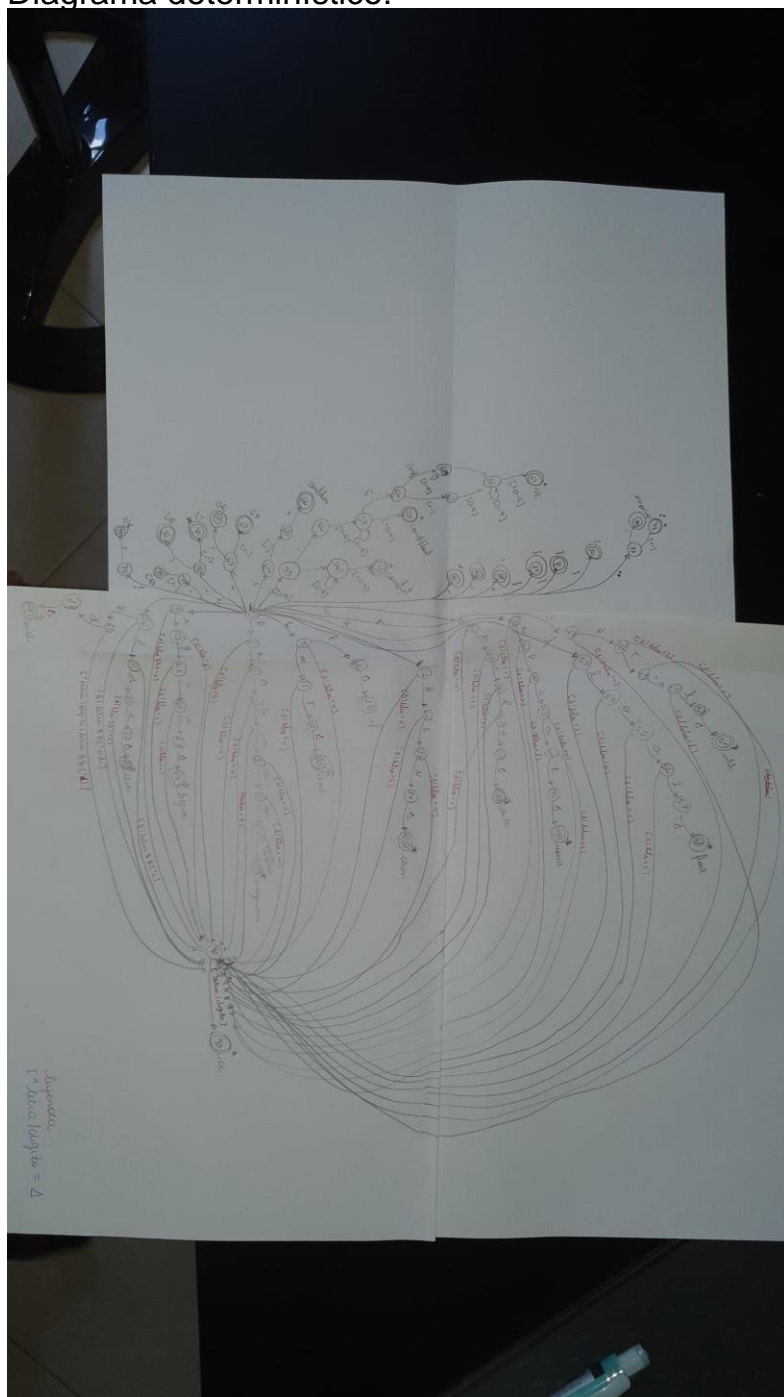


Diagrama determinístico:



Análise Léxica

3ª Etapa do Projeto

Análise sintática (especificação)

Fazer os ajustes necessários para que a GLC da linguagem seja LL(1)

S -> program id() Bloco

SequenciaComandos -> ComandoAtribuicao | ComandoRepeticao |
ComandoCondicional

Bloco -> begin DeclaracaoVariavel SequenciaComandos end

DeclaracaoVariavel -> Tipo: Lista_ids; | ϵ

Lista_ids -> id Lista_ids'
Lista_ids' -> , Lista_ids | ϵ

Tipo -> char | float | int

ComandoCondicional -> if Condição then ComandoBloco ComandoCondicional'
ComandoCondicional' -> else ComandoBloco | ϵ

ComandoRepeticao -> while Condição repeat ComandoBloco | repeat ComandoBloco
until Condição
ComandoBloco -> Bloco | SequenciaComandos

ComandoAtribuicao -> ID := ComandoAtribuicao'
ComandoAtribuicao' -> Expressao | constchar

Condição -> Expressao relop Expressao

Expressao -> Termo Expressao '
Expressao ' -> + Termo Expressao ' | - Termo Expressao ' | ϵ
Termo -> Fator Termo'
Termo' -> * Fator Termo' | / Fator Termo' | ϵ
Fator -> id | constint | constfloat | (Expressao)

Calcular FIRST e FOLLOW para os símbolos da gramática

Firsts:

$\text{FIRST}(\text{Fator}) = \{ \text{id}, \text{constint}, \text{constfloat}, (\}$
 $\text{FIRST}(\text{Termo}) = \{ \text{id}, \text{constint}, \text{constfloat}, (\}$
 $\text{FIRST}(\text{Termo}') = \{ *, /, \epsilon \}$
 $\text{FIRST}(S) = \{ \text{program} \}$
 $\text{FIRST}(\text{SequenciaComandos}) = \{ \text{id}, \text{if}, \text{while}, \text{repeat} \}$
 $\text{FIRST}(\text{Bloco}) = \{ \{ \}$
 $\text{FIRST}(\text{DeclaracaoVariavel}) = \{ \text{char}, \text{float}, \text{int} \}$
 $\text{FIRST}(\text{Lista_ids}) = \{ \text{id} \}$
 $\text{FIRST}(\text{Lista_ids}') = \{ , \}$
 $\text{FIRST}(\text{tipo}) = \{ \text{char}, \text{float}, \text{int} \}$
 $\text{FIRST}(\text{ComandoBloco}) = \{ \{, \text{id}, \text{if}, \text{while}, \text{repeat} \}$
 $\text{FIRST}(\text{ComandoCondicional}) = \{ \text{if} \}$
 $\text{FIRST}(\text{ComandoCondicional}') = \{ \text{else}, \epsilon \}$
 $\text{FIRST}(\text{ComandoRepeticao}) = \{ \text{while}, \text{repeat} \}$
 $\text{FIRST}(\text{ComandoAtribuicao}) = \{ \text{id} \}$
 $\text{FIRST}(\text{ComandoAtribuicao}') = \{ \text{id}, \text{constint}, \text{constfloat}, (\}$
 $\text{FIRST}(\text{Condição}) = \{ \text{id}, \text{constint}, \text{constfloat}, (\}$
 $\text{FIRST}(\text{Expressao}) = \{ \text{id}, \text{constint}, \text{constfloat}, (\}$
 $\text{FIRST}(\text{Expressao}') = \{ +, -, \epsilon \}$
 $\text{FIRST}(\text{ID}) = \{ \text{id} \}$
 $\text{FIRST}(\text{program}) = \{ \text{program} \}$
 $\text{FIRST}(\}) = \{ \}$
 $\text{FIRST}(\{) = \{ \{ \}$
 $\text{FIRST}(:) = \{ : \}$
 $\text{FIRST}(\text{INT}) = \{ \text{int} \}$
 $\text{FIRST}(\text{CHAR}) = \{ \text{char} \}$
 $\text{FIRST}(\text{FLOAT}) = \{ \text{float} \}$
 $\text{FIRST}(,) = \{ , \}$

$\text{FIRST}(;) = \{ ; \}$

$\text{FIRST}(\text{if}) = \{ \text{if} \}$

$\text{FIRST}(\text{then}) = \{ \text{then} \}$

$\text{FIRST}(\text{else}) = \{ \text{else} \}$

$\text{FIRST}(()) = \{ (\}$

$\text{FIRST}() = \{) \}$

$\text{FIRST}(\text{while}) = \{ \text{while} \}$

$\text{FIRST}(\text{repeat}) = \{ \text{repeat} \}$

$\text{FIRST}(\text{until}) = \{ \text{until} \}$

$\text{FIRST}(=) = \{ = \}$

$\text{FIRST}(<) = \{ < \}$

$\text{FIRST}(>) = \{ > \}$

$\text{FIRST}(<>) = \{ <> \}$

$\text{FIRST}(==) = \{ == \}$

$\text{FIRST}(<=) = \{ <= \}$

$\text{FIRST}(>=) = \{ >= \}$

$\text{FIRST}(+) = \{ + \}$

$\text{FIRST}(-) = \{ - \}$

$\text{FIRST}(*) = \{ * \}$

$\text{FIRST}(/) = \{ / \}$

$\text{FIRST}(\wedge) = \{ \wedge \}$

$\text{FIRST}(\text{constint}) = \{ \text{constint} \}$

$\text{FIRST}(\text{constfloat}) = \{ \text{constfloat} \}$

$\text{FIRST}(\text{constchar}) = \{ \text{constchar} \}$

$\text{FIRST}(\text{relop}) = \{ \text{relop} \}$

Follow:

$\text{FOLLOW}(S) = \{ \$ \}$

$\text{FOLLOW}(\text{sequenciaComandos}) = \{ '}' \}$

$\text{FOLLOW}(\text{bloco}) = \text{FIRST}(\text{ComandoCondicional}) - \{ \epsilon \} + \text{FOLLOW}(\text{ComandoBloco})$
 $\text{FOLLOW}(\text{ComandoCondicional}) + \text{FOLLOW}(\text{ComandoRepeticao}) = \{ \$, \text{until}, '}' \}$

$\text{FOLLOW}(\text{declaracaoVariavel}) = \text{FIRST}(\text{ComandoAtribuicao}) +$
 $\text{FIRST}(\text{ComandoCondicional}) + \text{FIRST}(\text{ComandoRepeticao}) = \{ \text{id, if, while, repeat} \}$

$\text{FOLLOW}(\text{Lista_ids}) = \text{FIRST}(\text{SequenciaComandos}) = \{ ; \}$

$\text{FOLLOW}(\text{Lista_ids}') = \text{FOLLOW}(\text{Lista_ids}) = \{ ; \}$

$\text{FOLLOW}(\text{Tipo}) = \{ : \}$

$\text{FOLLOW}(\text{ComandoCondicional}) = \text{FOLLOW}(\text{SequenciaComandos}) = \{ '}' \}$

$\text{FOLLOW}(\text{ComandoRepeticao}) = \text{FOLLOW}(\text{SequenciaComandos}) = \{ '}' \}$

$\text{FOLLOW}(\text{ComandoAtribuicao}) = \text{FOLLOW}(\text{SequenciaComandos}) = \{ '}' \}$

$\text{FOLLOW}(\text{condição}) = \{ '}' \}$

$\text{FOLLOW}(\text{Expressao}) = \{ '}', \text{relop}, '}' \}$

$\text{FOLLOW}(\text{Fator}) = \text{FIRST}(\text{Termo}') = \{ * \}$

$\text{FOLLOW}(\text{Termo}) = \text{FIRST}(\text{Expressao}') = \{ +, - \}$

$\text{FOLLOW}(\text{Termo}') = \text{FIRST}(\text{expres Expressao' sao'}) - \{ \epsilon \} + \text{FOLLOW}(\text{Termo}) = \{ +, - \}$

$\text{FOLLOW}(\text{ComandoCondicional}') = \text{FOLLOW}(\text{ComandoCondicional}) = \{ '}' \}$

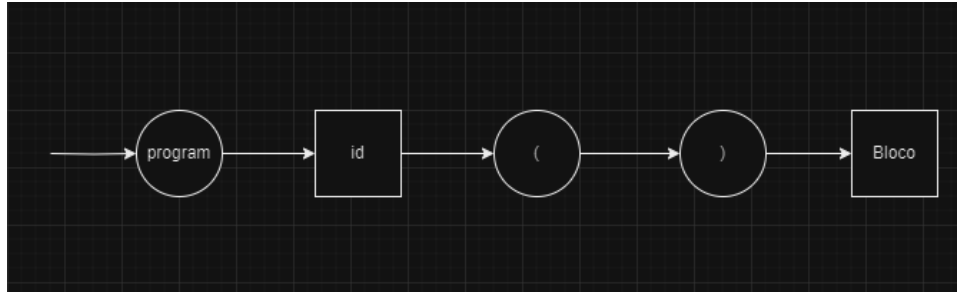
$\text{FOLLOW}(\text{Expressao}') = \{ + \}$

$\text{FOLLOW}(\text{ComandoAtribuicao}') = \text{FOLLOW}(\text{ComandoAtribuicao}) =$
 $\text{FOLLOW}(\text{SequenciaComandos}) = \{ '}' \}$

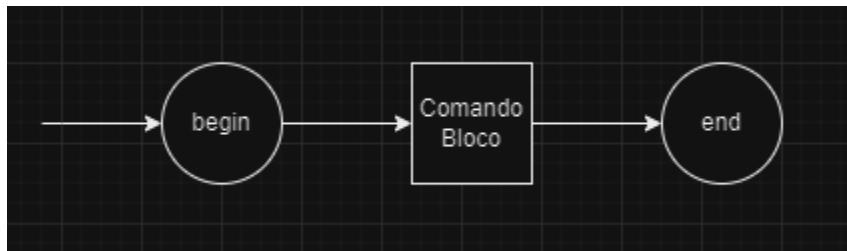
$\text{FIRST}(\text{ComandoBloco}) = \{ '{', \text{id, if, while, repeat} \}$

Grafos Sintáticos:

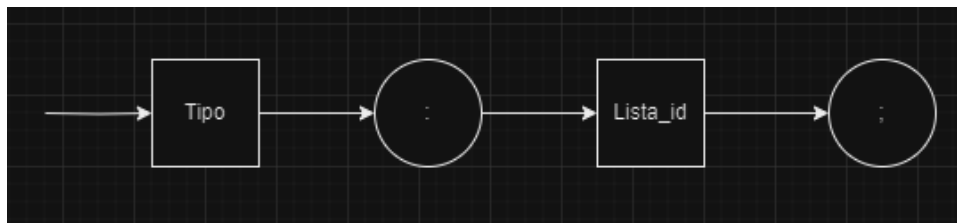
S



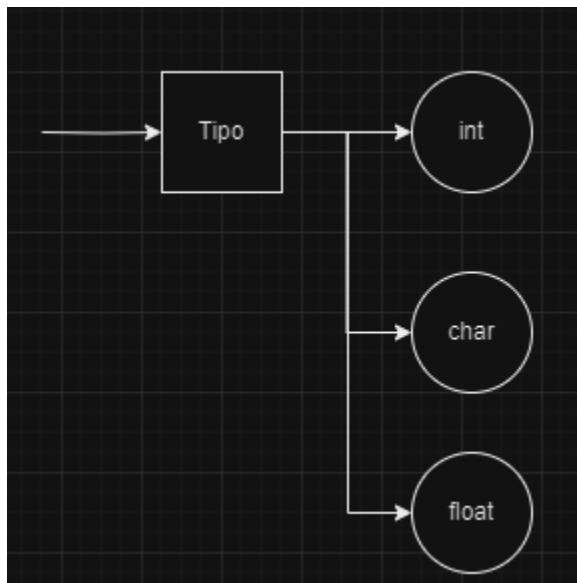
Bloco



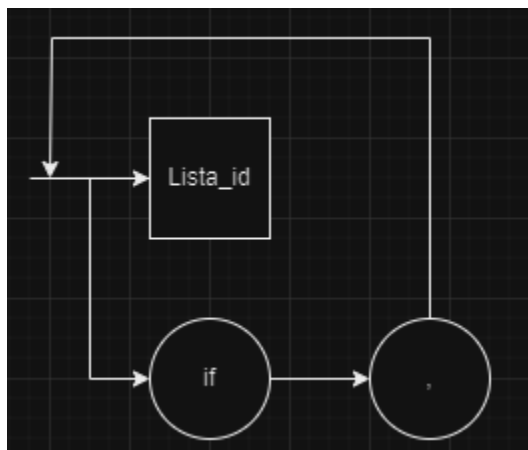
Declaração



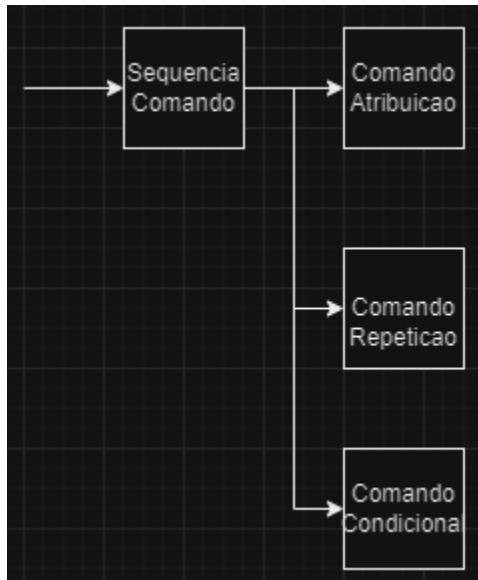
Tipo



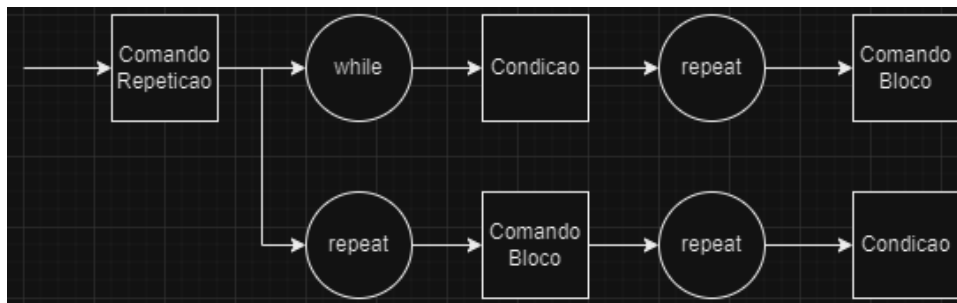
Lista_id



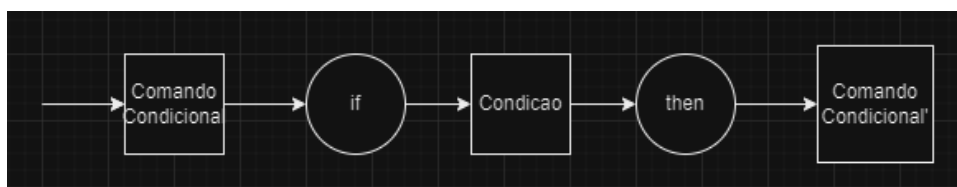
Sequência Comando



Comando Repetição



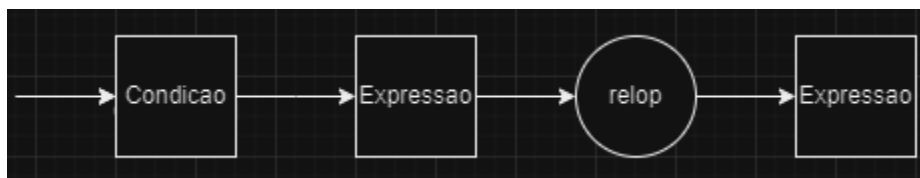
Comando Condicional



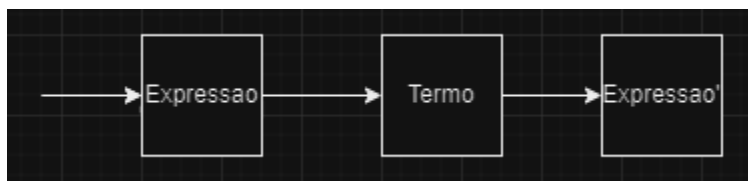
Comando Condicional'



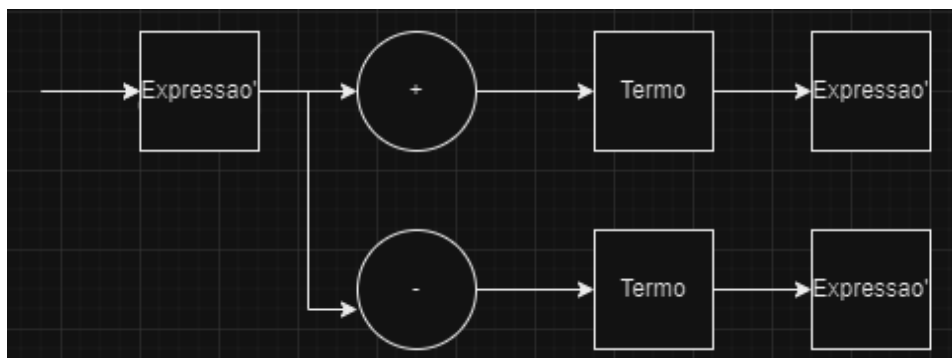
Condição



Expressão



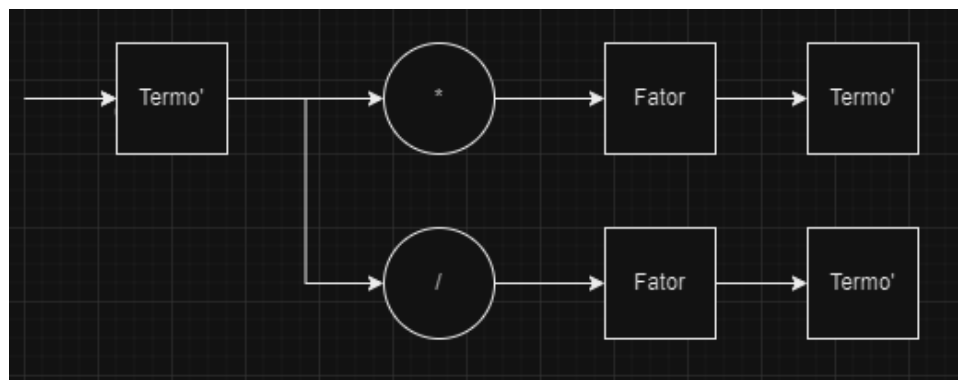
Expressão'



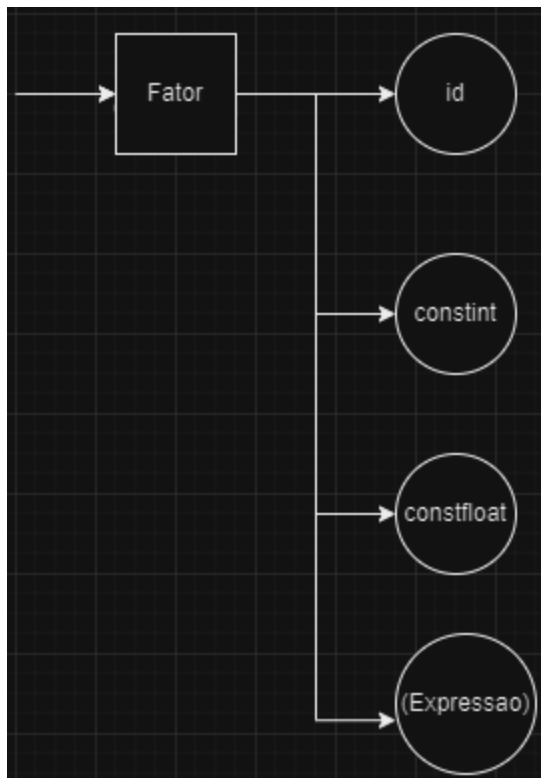
Termo



Termo'



Fator



Relop

