**Gramática do Projeto**

**Regra Inicial:**

t\_programa = "Program" t\_id t\_bloco

**Bloco de Código:**

t\_bloco = t\_abreBloco [ declaracao\_variavel ] [ comando ] t\_fechaBloco

**Declaração de Variável:**

declaracao\_variavel = tipo identificador\_inicializador { ',' identificador\_inicializador } t\_ponto\_virgula

tipo = t\_integer | t\_float | t\_char | t\_string | t\_boolean

identificador\_inicializador = t\_id [ t\_atribuicao t\_expr ]

**Comandos:**

comando = comando\_if | comando\_while | comando\_atribuicao | t\_bloco

comando\_atribuicao = t\_id t\_atribuicao t\_expr t\_ponto\_virgula

comando\_if = t\_if t\_abreParen t\_exprRelacional t\_fechaParen t\_bloco [ t\_else t\_bloco ]

comando\_while = t\_while t\_abreParen t\_exprRelacional t\_fechaParen t\_bloco

**Expressões:**

t\_exprRelacional = t\_expr operador\_relacional t\_expr | t\_expr | t\_bool

t\_expr = t\_termo { operador\_aditivo t\_termo }

t\_termo = t\_fator { operador\_multiplicativo t\_fator }

t\_fator = t\_id | t\_numero\_int | t\_numero\_real | t\_char\_literal | t\_string\_literal | t\_abreParen t\_expr t\_fechaParen

**Tokens de Tipos e Palavras-chave:**

t\_programa = "Program"

t\_integer = "Integer"

t\_float = "Float"

t\_char = "Char"

t\_string = "String"

t\_if = "If"

t\_else = "Else"

t\_while = "While"

t\_boolean = "Boolean"

**Tokens de Operadores:**

operador\_relacional = t\_igualdade | t\_diferenca | t\_menor | t\_maior | t\_menor\_igual | t\_maior\_igual

operador\_aditivo = t\_soma | t\_subtracao

operador\_multiplicativo = t\_multiplicacao | t\_divisao

t\_atribuicao = "="

t\_igualdade = "=="

t\_diferenca = "!="

t\_menor = "<"

t\_maior = ">"

t\_menor\_igual = "<="

t\_maior\_igual = ">="

t\_soma = "+"

t\_subtracao = "-"

t\_multiplicacao = "\*"

t\_divisao = "/"

**Tokens de Símbolos:**

t\_abreBloco = "{

t\_fechaBloco = "}"

t\_abreParen = "("

t\_fechaParen = ")"

t\_virgula = ","

t\_ponto = "."

t\_ponto\_virgula = ";"

**Tokens Básicos (Definição Lexical):**

t\_id = letra\_min { letra | digito }

t\_numero\_int = digito { digito }

t\_numero\_real = digito { digito } '.' digito { digito }

t\_char\_literal = " ' " letra " ' "

t\_string\_literal = ' " ' { letra } ' " '

t\_bool = "true" | "false"

letra = 'a'..'z' | 'A'..'Z'

letra\_min = 'a'..'z'

digito = '0'..'9'

**First:**

FIRST(t\_programa) = { "Program" }

FIRST(t\_defvar) = { "Integer", "Float", "Char", "String"}

FIRST(t\_comando) = { "If", "Else", "While", t\_id, "{" }

FIRST(t\_expr) = { t\_id, t\_num, "(" }

FIRST(comando\_if) = { "If" }

FIRST(comando\_else) = { "Else" }

FIRST(comando\_while) = { "While" }

FIRST(t\_exprRelacional) = { t\_id, t\_num, "(" }

FIRST(t\_oprel) = { "==", "!=", "<", ">", "<=", ">=" }

**Follow:**

FOLLOW(t\_programa) = { "$" }

FOLLOW(t\_comando) = { "}", "$" }

FOLLOW(t\_expr) = { ";", ")", "}" }

FOLLOW(comando\_if) = { "Else", FOLLOW(t\_comando) }

FOLLOW(comando\_while) = { FOLLOW(t\_comando) }

FOLLOW(t\_exprRelacional) = { ")",

FOLLOW(t\_comando) }

FOLLOW(t\_oprel) = { t\_id, t\_num, "(" }

**Delimitação de Tokens** Espaços em branco são usados como delimitadores de tokens. *(Isso é uma limitação da implementação atual do Analisador Léxico. Uma implementação robusta ignoraria espaços exceto como separadores)* A linguagem diferencia maiúsculas e minúsculas. Terminais são representados por palavras reservadas em MAIÚSCULAS ou símbolos entre aspas/apóstrofos. *(Nota: Atualizei a notação para usar aspas para literais de string)* Não terminais são representados em minúsculas.