

Gramática livre de contexto

- **programa** -> tipo 'charles()' '->' '{' bloco '}'
- **bloco** -> cmd bloco_options
- **bloco_options** -> bloco | e
- **cmd** -> leitura | escrita | atribuicao | cmd_if | declarar
- **declarar** -> tipo declarar_options
- **declarar_options** -> id ';' | atribuicao
- **cmd_if** -> 'if' '('condicional')' '->' '{' bloco '}' option_else
- **option_else** -> 'else' '{' bloco '}' | e
- **condicional** -> expressão operador_relacional expressão condicional' '!' expressão condicional'
- **condicional'** -> operador_logico condicional condicional'
- **expressão** -> exp_prioridade expressão'
- **expressão'** -> '+' exp_prioridade expressão' | - '-' exp_prioridade expressão' | e
- **exp_prioridade** -> fator exp_prioridade'
- **exp_prioridade'** -> '*' fator exp_prioridade' | '/' fator exp_prioridade' | e
- **fator** -> id | num | '(' expressão ')'
- **leitura** -> id '<<' 'input' ';'
- **escrita** -> 'console' '<<' escrita_options ';'
- **escrita_options** -> id | texto_string
- **atribuicao** -> id '<-' atribuicao_options ';'
- **atribuicao_options** -> expressão | texto_string
- **texto_string** -> ""[a-zA-Z 0-9]""
- **num** -> num_int | num_decimal
- **operador_relacional** -> '>' | '<' | '>=' | '<=' | '=' | '!='
- **tipo** -> 'int' | 'float' | 'boolean' | 'string'
- **operador_logico** -> '&&' | '||' | '!'
- **id** -> [a-zA-Z][a-zA-Z0-9]*
- **num_int** -> [0-9]+
- **num_decimal** -> [0-9]+.[0-9]+
- **loop_for** -> 'for' '(' declarar ';' condicional ';' atribuicao ')' '->' '{' bloco '}'
- **loop_while** -> 'while' '('condicional')' '->' '{' bloco '}'

Exemplo de código

```
int charles() -> {  
  
    int variavel;  
    variavel << input;  
    bool estaNoIntervalo;  
    if ( variavel > 10 && variavel < 30 ) -> {  
        estaNoIntervalo = true;  
    }  
    if (!estaNoIntervalo){  
        console << " Não está no intervalo " ;  
    }  
    if(variavel = 30){  
        console<<" A 8==D variavel é exatamente 30";  
    }  
  
}
```