**BNF da Linguagem Bacco**

**Programa e Bloco**

1. <programa> ::=

Programa1 <identificador> ;

<bloco>.

2. <bloco> ::=

[<parte de definições de constantes>]

[<parte de definições de tipos>]

[<parte de declarações de variáveis>]

[<parte de declarações de sub-rotinas>]

<comando composto>

**Declarações**

3. <parte de definições de constantes> :: =

constante <definição de constante> {; <definição de constante>}

4. <definição de constante> ::= <identificador> = <constante>

5. <constante> ::= [+|-] (<identificador> | <numero\_inteiro> )

6. <parte das definições de tipo> ::=

tipo <definição de tipo> {; <definição de tipo> }

7. <definição de tipo> ::= <identificador> = <tipo>

8. <tipo> ::=

<identificador> |

<numero> |

<grava> |

lista [<indice>] de <tipo> |

palavra [<número\_inteiro>]

9. <indice> ::= < número\_inteiro > .. < número\_inteiro >

10. <numero> ::= ( <lista de identificadores> )

11. <grava> ::= grave <lista de campos> fim

12. <lista de campos> ::= <lista de identificadores> :

<tipo> { ; <lista de identificadores> : <tipo> }

13. <parte de declarações de variáveis> ::=

var <declaração de variáveis>

{; <declaração de variáveis>};

14.<declaração de variáveis>::=

<lista de identificadores> : <tipo>

15. <lista de identificadores> ::= <identificador> {, <identificador>}

16. <parte de declarações de subrotinas> ::=

{<declaração de procedimento> ; | <declaração de função> ;}

17. <declaração de procedimento> ::=

procedimento <identificador> [<parâmetros formais>] ;

<bloco>

18. <declaração de função> ::=

funcao <identificador> [<parâmetros formais>] : <identificador> ; <bloco>

19. <parâmetros formais> ::=

( <seção de parâmetros formais> { ; <seção de parâmetros formais>} )

20. <seção de parâmetros formais> ::= [var] <lista de identificadores> : <identificador>

**Comandos**

21. <comando composto> ::= comece <comando> { ; <comando>} fim

22. <comado> ::= <atribuição>

| <chamada de procedimento>  leia/escreva

| <comando composto>

| <comando condicional 1>

| <comando condicional 2>

| <comando repetitivo 1>

| <comando repetitivo 2>

| <comando repetitivo 3>

23. <atribuição>::= <variável> := <expressão>

24. <chamada de procedimento> ::= <identificador> [ ( <lista de expressões> ) ]

25. <comando condicional 1> ::= se <expressão> faca <comando>

[senao <comando>]

26. <comando condicional 2> ::= caso <expressão> seja <elemento do case> { ; <elemento

do case> } fim

27. <elemento do case>::= <constante> { , <constante> } : <comando>

28. <comando repetitivo 1> ::= enquanto <expressão> faca <comando>

29. <comando repetitivo 2> : := <expressão> faca <comando>:= para <identificador>

30. <comando repetitivo 3> ::= repita <comando> { ; <comando>} ate <expressão>

**Expressões**

31. <expressão> ::= <expressão simples> [<relação> <expressão simples>]

32. <relação> ::= = | <> | < | <= | >= | >

33. <expressão simples> ::= [+ | -] <termo> {(+ | - | or) <termo>}

34. <termo> ::= <fator> {(\* | div | and | / ) <fator> }

35. <fator> ::= <variavel>

| <número\_inteiro>

| <número\_real>

| <string>

| <char>

| <chamada de função>

| ( <expressão> )

| not <fator>

36. <variável> ::= <identificador> [ ( [ <expressão> ] | .

<campo> ) ]

37. <lista de expressões> ::= <expressão> {, <expressão>}

38. <chamada de função> ::= <identificador> [ (<lista de

expressões> ) ]

39. <campo> ::= <identificador>

<keyword> ::= se | senao | fim | para | enquanto | repita | int | divis | const | var | lista | palavra | faca | ate | seja | leia | escreva | lista | palavra | grave | de