

BNF da Linguagem Pascal Simplificada

```

<programa> ::= program <identificador> <corpo> .
<corpo> ::= <declaração> <rotina> begin <sentenças> end |
           <rotina> begin <sentenças> end |
           begin <sentenças> end
<declaração> ::= var <declaraçãoVariável> <maisDeclaração>
<maisDeclaração> ::= ; <continuaDeclaração> | ;
<continuaDeclaração> ::= <declaraçãoVariável> <maisDeclaração>
<declaraçãoVariável> ::= <variáveis>: <tipoVariável>
<tipoVariável> ::= integer
<variáveis> ::= <identificador>, <variáveis> | <identificador>
<rotina> ::= <procedimento> | <função>
<procedimento> ::= procedure <identificador> <parâmetros> ; <corpo>; <rotina> |
                 procedure <identificador> ; <corpo>; <rotina> |
                 procedure <identificador> <parâmetros> ; <corpo>;
                 procedure <identificador> ; <corpo>;
<função> ::= function <identificador> <parâmetros> : <tipoFunção> ; <corpo>; <rotina> |
           function <identificador> : <tipoFunção> ; <corpo>; <rotina> |
           function <identificador> <parâmetros> : <tipoFunção> ; <corpo>;
           function <identificador> : <tipoFunção> ; <corpo>;
<parâmetros> ::= ( <listaParâmetros> )
<listaParâmetros> ::= <listaIdentificador> : <tipoVariável> <maisParâmetros> |
                     <listaIdentificador> : <tipoVariável> <maisParâmetros>
<maisParâmetros> ::= ; <listaParâmetros>
<listaIdentificador> ::= <identificador> <maisIdentificador> | <identificador>
<maisIdentificador> ::= , <listaIdentificador>
<tipoFunção> ::= integer
<sentenças> ::= <comando> <maisSentença>
<maisSentenças> ::= ; <sentença> | ;
<comando> ::= read ( <variávelLer> ) | write ( <variávelEscreve> ) |
             for <identificador2> := <expressão> to <expressão> do begin <sentenças> end |
             repeat <sentenças> until ( <condição> ) |
             if ( <condição> ) then begin <sentenças> end |
             if ( <condição> ) then begin <sentenças> end else begin <sentença> end |
             <identificador2> := <expressão> | <chamada>
<variávelLer> ::= <identificador> , <variávelLer> | <identificador>
<variávelEscreve> ::= <identificador> , <variávelEscreve> | <identificador>
<chamada> ::= <identificador> <argumentos> | <identificador>
<argumentos> ::= ( <listaArgumento> )
<listaArgumentos> ::= <expressão> , <listaArgumento> | <expressão>
<condição> ::= <expressão> <relação> <expressão>
<relação> ::= = | > | < | >= | <= | <>
<expressão> ::= <termo> <maisExpressão>
<maisExpressão> ::= <operador1> <termo> <maisExpressão> | <operador> <termo>
<operador1> ::= + | -
<termo> ::= <fator> <maisFator>
<fator> ::= <identificador> | <numeroInteiro> | ( <expressão> ) | <identificador> <argumentos>
<maisFator> ::= <operador2> <fator> <maisFatores> | <operador2> <fator>
<operador2> ::= * | /
<numeroInteiro> ::= <digito> <numeroInteiro> | <digito>
<identificador> ::= <letra> <listaLetraDigito> | <letra>
<listaLetraDigito> ::= <letra> <listaLetraDigito> | <digito> <listaLetraDigito> | <letra> | <digito>
<letra> ::= a | b | c | d | e | f | g | h | i | j | l | m | n | o | p | q | r | s | t | u | v | x | z
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

BNF da Linguagem C Simplificada

```
<programa> ::= <rotina> main () { <corpo> } | main () { <corpo> }
<corpo> ::= <declaração> <sentenças>
<declaração> ::= <tipoVariável> <variável> ; <declaração> | <tipoVariável> <variável> ;
<tipoVariável> ::= int
<variável> ::= <identificador> , <variável> | <identificador>
<rotina> ::= <tipoFunção> <identificador> <parâmetros> { <corpo> } <rotina> |
           <tipoFunção> <identificador> { <corpo> } <rotina> |
           <tipoFunção> <identificador> <parâmetros> { <corpo> } |
           <tipoFunção> <identificador> { <corpo> } |
<parâmetros> ::= ( <listaParâmetros> )
<listaParâmetros> ::= <tipoVariável> <listaIdentificador> ; <listaParâmetros> |
                    <tipoVariável> <listaIdentificador> ;
<listaIdentificador> ::= <identificador> , <listaIdentificador> | <identificador>
<tipoFunção> ::= void | integer
<sentença> ::= <simples> ; <sentença> | <comando> <sentença> | <simples> ;
<simples> ::= printf ( <variávelEscreve> )          | scanf ( <variávelLer> ) |
              <identificador> = <expressão>          | <identificador> <argumentos>
<variávelEscreve> ::= <identificador> , <variávelEscreve> | <identificador>
<variávelLer> ::= <identificador> , <variávelLer> | <identificador>
<argumentos> ::= ( <listaArgumento> )
<listaArgumento> ::= <expressão> , <listaArgumento> | <expressão>
<comando> ::= do <sentença> while ( <condição> ) |
              while ( <condição> ) { <sentença> } |
              if ( <condição> ) { <sentença> } else { <sentença> } |
              if ( <condição> ) { <sentença> } |
              for ( <identificador> = <expressão> ; <condição> ; <identificador> = <expressão> ) { <sentença> }
<condição> ::= <expressão> <relação> <expressão>
<relação> ::= == | > | < | >= | <= | !=
<expressão> ::= <termo> <maisExpressão>
<maisExpressão> ::= <operador1> <termo> <maisExpressão> | <operador> <termo>
<operador1> ::= + | -
<termo> ::= <fator> <maisFator>
<fator> ::= <identificador> | <numeroInteiro> | ( <expressão> ) | <identificador> <argumentos>
<maisFator> ::= <operador2> <fator> <maisFatores> | <operador2> <fator>
<operador2> ::= * | /
<numeroInteiro> ::= <digito> <numeroInteiro> | <digito>
<identificador> ::= <letra> <listaLetraDigito> | <letra>
<listaLetraDigito> ::= <letra> <listaLetraDigito> | <digito> <listaLetraDigito> | <letra> | <digito>
<letra> ::= a | b | c | d | e | f | g | h | i | j | l | m | n | o | p | q | r | s | t | u | v | x | z
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```