Gramática

<programa></programa>	<pre><def_tipos> <def_variaveis> <def_funcoes></def_funcoes></def_variaveis></def_tipos></pre>										
<def_tipos></def_tipos>	<def_< td=""><td>TIPO></td><td>;</td><td><def_< td=""><td>TIPOS></td><td></td><td></td><td></td><td></td><td></td><td></td></def_<></td></def_<>	TIPO>	;	<def_< td=""><td>TIPOS></td><td></td><td></td><td></td><td></td><td></td><td></td></def_<>	TIPOS>						
	€										
<def_variaveis></def_variaveis>	var	<tipo_< td=""><td>_DADO></td><td>•</td><td><id></id></td><td><def_< td=""><td>VAR></td><td><def_< td=""><td>VARIAVI</td><td>EIS></td><td></td></def_<></td></def_<></td></tipo_<>	_DADO>	•	<id></id>	<def_< td=""><td>VAR></td><td><def_< td=""><td>VARIAVI</td><td>EIS></td><td></td></def_<></td></def_<>	VAR>	<def_< td=""><td>VARIAVI</td><td>EIS></td><td></td></def_<>	VARIAVI	EIS>	
	€										
<def_funcoes></def_funcoes>	functio	n	<func< td=""><td>AO></td><td><def_< td=""><td>FUNCO</td><td>ES></td><td></td><td></td><td></td><td></td></def_<></td></func<>	AO>	<def_< td=""><td>FUNCO</td><td>ES></td><td></td><td></td><td></td><td></td></def_<>	FUNCO	ES>				
	ϵ										
<def_tipo></def_tipo>	typede	f	<tipo_< td=""><td>_DADO></td><td></td><td><id></id></td><td></td><td></td><td></td><td></td><td></td></tipo_<>	_DADO>		<id></id>					
< DEF_VAR >	[<nume< td=""><td>RO></td><td>]</td><td>< DEF</td><td>_VAR ></td><td></td><td></td><td></td><td></td><td></td></nume<>	RO>]	< DEF	_VAR >					
	,	<id></id>	< DEF_	_VAR >							
	;										
<tipo_dado></tipo_dado>	int										
	float										
	Double)									
	struct		<id></id>	{	<def_< td=""><td>VARIAV</td><td>EIS></td><td>}</td><td></td><td></td><td></td></def_<>	VARIAV	EIS>	}			
	<id></id>										
<funcao></funcao>	<tipo_< td=""><td>_DADO></td><td>•</td><td><id></id></td><td>(</td><td>< DEF_</td><td>VARIA</td><td>VEIS ></td><td></td><td>)</td><td><bloco></bloco></td></tipo_<>	_DADO>	•	<id></id>	(< DEF_	VARIA	VEIS >)	<bloco></bloco>
<bloco></bloco>	{	<def_< td=""><td>VARIA\</td><td>/EIS ></td><td><com< td=""><td>ANDOS></td><td>•</td><td>}</td><td></td><td></td><td></td></com<></td></def_<>	VARIA\	/EIS >	<com< td=""><td>ANDOS></td><td>•</td><td>}</td><td></td><td></td><td></td></com<>	ANDOS>	•	}			
COMANDOO:											
<comandos></comandos>	<com <="" td=""><td>ANDO></td><td>;</td><td><coma< td=""><td>NDOS</td><td>></td><td></td><td></td><td></td><td></td><td></td></coma<></td></com>	ANDO>	;	<coma< td=""><td>NDOS</td><td>></td><td></td><td></td><td></td><td></td><td></td></coma<>	NDOS	>					
<comandos></comandos>	<com <="" td=""><td>ANDO></td><td>;</td><td><coma< td=""><td>NDOS</td><td>></td><td></td><td></td><td></td><td></td><td></td></coma<></td></com>	ANDO>	;	<coma< td=""><td>NDOS</td><td>></td><td></td><td></td><td></td><td></td><td></td></coma<>	NDOS	>					
<comandos></comandos>		ANDO> <nome< td=""><td></td><td><com<i>A</com<i></td><td>NDOS:</td><td></td><td></td><td></td><td></td><td></td><td></td></nome<>		<com<i>A</com<i>	NDOS:						
	€		≣>		<valc< td=""><td></td><td><blo0< td=""><td>00></td><td></td><td></td><td></td></blo0<></td></valc<>		<blo0< td=""><td>00></td><td></td><td></td><td></td></blo0<>	00>			
	€ <id></id>		=> <exp_< td=""><td>=</td><td><valc< td=""><td>DR></td><td><blo0< td=""><td></td><td><else< td=""><td>></td><td></td></else<></td></blo0<></td></valc<></td></exp_<>	=	<valc< td=""><td>DR></td><td><blo0< td=""><td></td><td><else< td=""><td>></td><td></td></else<></td></blo0<></td></valc<>	DR>	<blo0< td=""><td></td><td><else< td=""><td>></td><td></td></else<></td></blo0<>		<else< td=""><td>></td><td></td></else<>	>	
	€ <id> while</id>	<nome (<="" td=""><td>=> <exp_ <exp_< td=""><td>= LOGICA</td><td><valc > ></valc </td><td>))</td><td></td><td></td><td><else< td=""><td>></td><td></td></else<></td></exp_<></exp_ </td></nome>	=> <exp_ <exp_< td=""><td>= LOGICA</td><td><valc > ></valc </td><td>))</td><td></td><td></td><td><else< td=""><td>></td><td></td></else<></td></exp_<></exp_ 	= LOGICA	<valc > ></valc))			<else< td=""><td>></td><td></td></else<>	>	
	€ <id> while</id>	<nome (="" (<="" th=""><th>=> <exp_ <exp_< th=""><th>= LOGICA LOGICA</th><th><valc > ></valc </th><th>))</th><th></th><th></th><th><else< th=""><th>></th><th></th></else<></th></exp_<></exp_ </th></nome>	=> <exp_ <exp_< th=""><th>= LOGICA LOGICA</th><th><valc > ></valc </th><th>))</th><th></th><th></th><th><else< th=""><th>></th><th></th></else<></th></exp_<></exp_ 	= LOGICA LOGICA	<valc > ></valc))			<else< th=""><th>></th><th></th></else<>	>	
	€ <id> while if printf scanf</id>	<nome (="" (<="" td=""><td>E> <exp_ <exp_="" <id="" <nome=""></exp_></td><td>= LOGICA LOGICA E_NUME</td><td><valc > ></valc </td><td>)))</td><td></td><td></td><td><else></else></td><td>></td><td></td></nome>	E> <exp_ <exp_="" <id="" <nome=""></exp_>	= LOGICA LOGICA E_NUME	<valc > ></valc)))			<else></else>	>	
	€ <id> while if printf scanf</id>	<nome (="" (<="" td=""><td>E> <exp_ <exp_="" <id="" <nome=""> R></exp_></td><td>= LOGICA LOGICA E_NUME</td><td><valc > ></valc </td><td>)))</td><td></td><td></td><td><else< td=""><td>></td><td></td></else<></td></nome>	E> <exp_ <exp_="" <id="" <nome=""> R></exp_>	= LOGICA LOGICA E_NUME	<valc > ></valc)))			<else< td=""><td>></td><td></td></else<>	>	
<comando></comando>	€ <id> while if printf scanf return</id>	<nome (="" <="" td="" valo<=""><td>E> <exp_ <exp_="" <id="" <nome=""> R></exp_></td><td>= LOGICA LOGICA E_NUME</td><td><valc > ></valc </td><td>)))</td><td></td><td></td><td><else< td=""><td>></td><td></td></else<></td></nome>	E> <exp_ <exp_="" <id="" <nome=""> R></exp_>	= LOGICA LOGICA E_NUME	<valc > ></valc)))			<else< td=""><td>></td><td></td></else<>	>	
<comando></comando>	ϵ <id> while if printf scanf return else ϵ</id>	<nome (="" <="" td="" valo<=""><td>E> <exp_ <exp_="" <id="" <nome=""> R> O></exp_></td><td>= LOGICA LOGICA E_NUME</td><td><valc > ></valc </td><td>)))</td><td></td><td></td><td><else< td=""><td>></td><td></td></else<></td></nome>	E> <exp_ <exp_="" <id="" <nome=""> R> O></exp_>	= LOGICA LOGICA E_NUME	<valc > ></valc)))			<else< td=""><td>></td><td></td></else<>	>	

<parametros></parametros>	(<parametro>)</parametro>						
<parametro></parametro>	<lista_param></lista_param>						
	€						
<lista_param></lista_param>	<nome_numero> , <lista_param></lista_param></nome_numero>						
	<nome_numero></nome_numero>						
<exp_logica></exp_logica>	<exp_matematica> <op_logico> <exp_logica></exp_logica></op_logico></exp_matematica>						
	<exp_matematica></exp_matematica>						
<op_logico></op_logico>	> < == !=						
<exp_matematica></exp_matematica>	<nome_numero> <op_ matematico=""> <exp_ matematica=""></exp_></op_></nome_numero>						
	<nome_numero></nome_numero>						
<op_matematico></op_matematico>	+ - * /						
<nome_numero></nome_numero>	<id> <nome></nome></id>						
	<numero></numero>						
<nome></nome>	. <id> <nome></nome></id>						
	[<nome_numero>]</nome_numero>						
	(<parametro>)</parametro>						
	ϵ						
<id></id>	Seqüência alfanumérica iniciada por char (tratado no lexico)						
<numero></numero>	seqüência numérica com no máximo um ponto						

```
typedef double duplo;
typedef struct al {
                 var float nota1, nota2;
         } aluno;
var int A, B, C, D;
var duplo E[15];
var aluno F;
fuction int fatorial(var int a;){
   var int i,result;
  i = 1;
  result =1;
  while (i < a) {
      result =result*i;
      i:=i+1;
   };
   return result;
}
function float exp(var float a, b;){
   var int i:
   var float result;
   i = 1;
   result = a;
   if (b == 0) {
         result = 1;
   } else {
         while (i < b)
                 result = a * a;
                 i = i + 1;
        };
   };
   return result;
}
function double maior(var vetor a;){
   var int i:
   var double result;
   i = 0;
   result = a[0];
   while (i < 15){
         if (a[i] > result) {
            result = a[i];
        };
   };
    return result;
}
```

```
function aluno lerDados(){
    var aluno result;
    var string msg;
    msg = "digite as notas do aluno";
    printf(msg);
    scanf(result.nota1);
    scanf(result.nota2);
    return result;
}

function int main(){
    A = 10;
    B = fatorial(A);
    C = exp(A,B);
    D = maior(E);
}
```