

## Gramática

<PROGRAMA>	<DEF_TIPOS> <DEF_VARIAVEIS> <DEF_FUNCOES>
<DEF_TIPOS>	<DEF_TIPO> ; <DEF_TIPOS>
	€
<DEF_VARIAVEIS>	<b>var</b> <TIPO_DADO> <ID> <DEF_VAR> <DEF_VARIAVEIS>
	€
<DEF_FUNCOES>	<b>function</b> <FUNCAO> <DEF_FUNCOES>
	€
<DEF_TIPO>	<b>typedef</b> <TIPO_DADO> <ID>
<DEF_VAR>	[ <NUMERO> ] <DEF_VAR>
	, <ID> <DEF_VAR>
	;
<TIPO_DADO>	<b>int</b>
	<b>float</b>
	<b>Double</b>
	<b>struct</b> <ID> { <DEF_VARIAVEIS> }
	<ID>
<FUNCAO>	<TIPO_DADO> <ID> ( <DEF_VARIAVEIS> ) <BLOCO>
<BLOCO>	{ <DEF_VARIAVEIS> <COMANDOS> }
<COMANDOS>	<COMANDO> ; <COMANDOS>
	€
<COMANDO>	<ID> <NOME> = <VALOR>
	<b>while</b> ( <EXP_LOGICA> ) <BLOCO>
	<b>if</b> ( <EXP_LOGICA> ) <BLOCO> <ELSE>
	<b>printf</b> ( <NOME_NUMERO> )
	<b>scanf</b> ( <ID> <NOME> )
	<b>return</b> <VALOR>
<ELSE>	<b>else</b> <BLOCO>
	€
<VALOR>	<EXP_MATEMATICA>
	“ sequencia alfanumérica ”

<PARAMETROS>	(      <PARAMETRO>      )
<PARAMETRO>	<LISTA_PARAM>
	€
<LISTA_PARAM>	<NOME_NUMERO>      ,      <LISTA_PARAM>
	<NOME_NUMERO>
<EXP_LOGICA>	<EXP_MATEMATICA>      <OP_LOGICO>      <EXP_LOGICA>
	<EXP_MATEMATICA>
<OP_LOGICO>	>             <             ==             !=
<EXP_MATEMATICA>	<NOME_NUMERO>      <OP_MATEMATICO>      <EXP_MATEMATICA>
	<NOME_NUMERO>
<OP_MATEMATICO>	+             -             *             /
<NOME_NUMERO>	<ID>      <NOME>
	<NUMERO>
<NOME>	.      <ID>      <NOME>
	[      <NOME_NUMERO>      ]
	(      <PARAMETRO>      )
	€
<ID>	Seqüência alfanumérica iniciada por char (tratado no lexico)
<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);
}

```