

Gramática Pyscal

[PROGRAMA]	[DECLARACOES]	[BLOCO]
[DECLARACOES]	[DEF_CONST]	[DEF_TIPOS] [DEF_VAR] [DEF_ROT]
[DEF_CONST]	(const) [CONSTANTE] (;) [LIST_CONST] ϵ	
[LIST_CONST]	[CONSTANTE] (;) [LIST_CONST] ϵ	
[CONSTANTE]	[ID] (=) [CONST_VALOR]	
[CONST_VALOR]	Seqüência alfanumérica iniciada por aspas e terminada em aspas (tratado no léxico) [EXP_MAT]	
[DEF_TIPOS]	(type) [TIPO] (;) [LIST_TIPOS] ϵ	
[LIST_TIPOS]	[TIPO] (;) [LIST_TIPOS] ϵ	
[TIPO]	[ID] (=) [TIPO_DADO]	
[TIPO_DADO]	(integer)	
	(real)	
	(array) (()) [NUMERO] (()) (of) [TIPO_DADO]	
	(record) [CAMPOS] (end)	
	[ID]	
[CAMPOS]	[ID] (;) [TIPO_DADO] [LISTA_CAMPOS]	
[LISTA_CAMPOS]	(;) [CAMPOS] [LISTA_CAMPOS] ϵ	
[DEF_VAR]	(var) [VARIABEL] (;) [LIST_VAR] ϵ	
[LIST_VAR]	[VARIABEL] (;) [LIST_VAR] ϵ	
[VARIABEL]	[ID] [LISTA_ID] (;) [TIPO_DADO]	
[LISTA_ID]	(,) [ID] [LISTA_ID] ϵ	
[DEF_ROT]	[NOME_ROTINA] [DEF_VAR] [BLOCO] [DEF_ROT]	
	ϵ	
[NOME_ROTINA]	(function) [ID] [PARAM_ROT] (;) [TIPO_DADO] (procedure) [ID] [PARAM_ROT]	
[PARAM_ROT]	(()) [CAMPOS] (())	
	ϵ	
[BLOCO]	(begin) [COMANDO] (;) [LISTA_COM] (end) (;) [COMANDO]	
[LISTA_COM]	[COMANDO] (;) [LISTA_COM]	
	ϵ	
[COMANDO]	[ID] [NOME] [ATRIB]	
	(while) [EXP_LOGICA] (do) [BLOCO]	
	(if) [EXP_LOGICA] (then) [BLOCO] [ELSE]	
	(return) [EXP_LOGICA]	
	(write) [EXP_MAT]	
	(read) [ID] [NOME]	
[ATRIB]	(:=) [EXP_MAT] ϵ	
[ELSE]	(else) [BLOCO] ϵ	
[LISTA_PARAM]	[PARAMETRO] (,) [LISTA_PARAM]	
	[PARAMETRO]	
	ϵ	
[OP_LOGICO]	(>) (<) (=) (!)	

[EXP_LOGICA]	[EXP_MAT]		[OP_LOGICO]	[EXP_LOGICA]
	[EXP_MAT]			
[EXP_MAT]	[PARAMETRO]		[OP_MAT]	[EXP_MAT]
	[PARAMETRO]			
[OP_MAT]	(+)	(-)	(*)	(/)
[PARAMETRO]	[ID]		[NOME]	
	[NUMERO]			
[NOME]	(.)		[ID]	[NOME]
	(I)	[PARAMETRO]		(I)
	(I)	[LISTA_PARAM]		(I)
	€			
[ID]	Seqüência alfanumérica iniciada por char (tratado no léxico)			
[NUMERO]	Seqüência numérica com a ocorrência de no máximo um ponto (tratado no léxico)			

```
const  TAM == 10;
      MSG == "digite as notas do aluno";
```

```
type   vetor == array [15] of integer;
      aluno == record
```

```
          nota1 : real;
          nota2 : real
```

```
      end;
```

```
var    A, B, C, D : integer;
      E : vetor;
      F : aluno;
```

```
procedure lerDados
begin
```

```
    write MSG;
    read F.nota1;
    read F.nota2;
```

```
end
```

```
function fatorial(a:integer) : integer
```

```
var i,result : integer;
begin
```

```
    i := 1;
    result:=1;
    while i < a do
    begin
        result:=result*i;
        i:=i+1;
    end;
    return result;
```

```
end
```

```
function exp(a: real; b: real) : real
```

```
var i,result : integer;
begin
```

```
    i := 1;
    result := a;
    if b = 0 then : result := 1
    else :   while i < b do
              begin
                  result := a * a;
                  i := i + 1;
              end;
    return result;
```

```
end
```

```
function maior(a : vetor) : integer
```

```
var i : integer;
```

```
begin
```

```
    i := 0;
    result := a[0];
    while i < 15 do
    begin
        if a[i] > result then : result := a[i];
        i := i + 1;
```

```
    end;
```

```
    return result;
```

```
end
```

```
function menor(a : vetor) : integer
```

```
var l, result : integer;
```

```
begin
```

```
    i := 0;
    result := a[0];
    while i < 15 do
    begin
        if a[i] < result then : result := a[i];
        i := i + 1;
```

```
    end
```

```
    return result;
```

```
end
```

```
function media(a : vetor) : integer
```

```
var m : integer;
```

```
begin
```

```
    m := maior(a) + menor(a);
    return m / 2;
```

```
end
```

```
begin
```

```
    A:=TAM + 20;
    B := fatorial(A);
    C := exp(A,B);
    D := media(E);
    lerDados();
```

```
end
```