

## First e Follow da Linguagem PasC

	<b>First</b>	<b>Follow</b>
<b>prog</b>	program	\$
<b>body</b>	num, char, {	\$
<b>decl-list</b>	num, char , $\epsilon$	{
<b>decl</b>	num, char	;
<b>type</b>	num, char	id
<b>id-list</b>	id	;
<b>id-list'</b>	, , $\epsilon$	;
<b>stmt-list</b>	id, if, while, read, write, $\epsilon$	}
<b>stmt</b>	id, if, while, read, write	;
<b>assign-stmt</b>	id	;
<b>if-stmt</b>	if	;
<b>if-stmt'</b>	else, $\epsilon$	;
<b>condition</b>	id, num_const, char_const, (, not	)
<b>while-stmt</b>	while	;
<b>stmt-prefix</b>	while	{
<b>read-stmt</b>	read	;
<b>write-stmt</b>	write	;
<b>writable</b>	id, num_const, char_const, (, not, literal	;
<b>expression</b>	id, num_const, char_const, (, not	)
<b>expression'</b>	==, >, >=, <, <=, !=, $\epsilon$	)
<b>simple-expr</b>	id, num_const, char_const, (, not	; , ==, >, >=, <, <=, !=, )
<b>simple-expr'</b>	+, -, or, $\epsilon$	; , ==, >, >=, <, <=, !=, )
<b>term</b>	id, num_const, char_const, (, not	+, -, or, ; , ==, >, >=, <, <=, !=, )
<b>term'</b>	*, /, and, $\epsilon$	+, -, or, ; , ==, >, >=, <, <=, !=, )
<b>factor-a</b>	id, num_const, char_const, (, not	*, /, and, +, -, or, ; , ==, >, >=, <, <=, !=, )
<b>factor</b>	id, num_const, char_const, (	*, /, and, +, -, or, ; , ==, >, >=, <, <=, !=, )
<b>relop</b>	==, >, >=, <, <=, !=	id, num_const, char_const, (, not
<b>addop</b>	+, -, or	id, num_const, char_const, (, not
<b>mulop</b>	*, /, and	id, num_const, char_const, (, not
<b>constant</b>	num_const, char_const	*, /, and, +, -, or, ; , ==, >, >=, <, <=, !=, )