

ERNESTO yUITI SAITO

vinicius tadeu gonçalves

gramática c

**LONDRINA - PR**

**2011**

INICIO -> DECLARAÇÃO; / FUNCAO / DECLARAÇÃO; INICIO / FUNÇÃO INICIO

DECLARAÇÃO -> TIPOPONTEIRO LISTAPONTEIRO / TIPOSEMPONTEIRO LISTAID

LISTA PONTEIRO ID-> id / id, LISTA ID

LISTA ID-> id / \*id / id, LISTA ID / \*id, LISTA ID

FUNÇÃO -> TIPOFUNÇÃO ID (DECLARAÇÃOFUNÇÃO) DECLARAÇÃOFUNCAOESPECIAL ESCOPOFUNCAO

ESCOPOFUNCAO -> {LISTAOPERAÇÕES} / ;

ESCOPO -> {LISTAOPERAÇÕES} / OPERAÇÃO

DECLARAÇÃOFUNÇÃO -> TIPOFUNÇÃO id / TIPOFUNÇÃO id , DECLARAÇÃOFUNÇÃO / NULL

DECLARAÇÃOFUNÇÃOESPECIAL -> TIPOFUNÇÃO id; / TIPOFUNÇÃO id ; DECLARAÇÃOFUNÇÃO / NULL

TIPOPONTEIRO -> char\* /int\* / char \*/int \*

TIPOSEMPONTEIRO -> char / int

TIPOFUNÇÃO -> TIPOPONTEIRO / TIPOSEMPONTEIRO / void / char[] / int[] / NULL

LISTAOPERAÇÕES -> OPERAÇÃO LISTA OPERAÇOES / OPERAÇÃO

OPERAÇÂO-> CONDICIONAL / DECLAÇÃO; / return VALOR ; / LOOP / EXPRESSÃO / ATRIBUIÇÃO / CHAMADAFUNCAO;

CHAMADAFUNCAO -> id(LISTA ID);

ATRIBUIÇÃO -> id = EXPRESSÃO

CONDICIONAL -> IF(LISTAEXPRESSÕES) ESCOPO / IF(LISTAEXPRESSÕES) ESCOPO ELSE ESCOPO / IF(LISTAEXPRESSÕES) OPERAÇÃO / IF(LISTAEXPRESSÕES) OPERAÇÃO ELSE ESCOPO / IF(LISTAEXPRESSÕES) ESCOPO ELSE OPERAÇÃO / IF(LISTAEXPRESSÕES) OPERAÇÃO ELSE OPERAÇÃO

VALOR -> id; / constante; / ; / EXPRESSÃO;

LISTAEXPRESSÕES -> EXPRESSÃO || LISTAEXPRESSõES / EXPRESSÃO && LISTA EXPRESSÕES / EXPRESSÃO / EXPRESSÃO / !(LISTAEXPRESSÔES) / !LISTAEXPRESSõES

EXPRESSÃO-> id OPERADOR id; / id = EXPRESSÃO / id; / vazio;

LOOP -> DO ESCOPO WHILE(EXPRESSÃO) / WHILE(EXPRESSÃO) ESCOPO / FOR(EXPRESSÃO; EXPRESSÃO; EXPRESSÃO) ESCOPO

OPERADOR -> = / % | / |