Gramatica Practica 2

Terminales

- Int
- float
- bool
- char*
- return
- if
- else
- while
- do
- print
- main
- +
- -
- *
- /
- %
- <
- >
- <=
- >=
- ==
- !=
- =
- &&,
- ||

- -=
- +=
- ++
- –-
- (
-)
- {
- }
- ;
- ,
- numero
- identificador
- cadena
- true
- false

No Terminales

- INICIO
- DECLARACION
- LISTA_VARS
- EXPRESION_LOGICA
- EXPRESION_RELACIONAL
- EXPRESION
- METODO
- LLAMADA
- LISTA_PARAMETROS
- PARAMETRO
- CONDICION
- ACCION

- LISTA_ACCIONES
- TIPO_DATO
- OPCIONES_DECLARACION
- OPCION_DOS
- OPERADOR_RELACIONAL
- ASIGNADORES
- LISTA_SENTENCIAS, SENTENCIA
- ASIGNACION_VAR
- WHILE
- DO_WHILE
- IF_ELSE
- DINCREMENTOS
- FUNCION_IMPRIMIR
- RETORNO
- ELSE
- PARS_LLAMADA
- PRINCIPAL
- SIMPLE.

Gramatica

- INICIO::= LISTA_ACCIONES | Epsilum
- LISTA_ACCIONES::= LISTA_ACCIONES ACCION | ACCION
- ACCION::= DECLARACION | METODO | FUNCION_IMPRIMIR | PRINCIPAL | ASIGNACION_VAR | DINCREMENTOS
- DECLARACION::= TIPO_DATO OPCIONES_DECLARACION
- OPCIONES_DECLARACION::= LISTA_VARS ; | OPCION_DOS ;
- LISTA_VARS::= LISTA_VARS , identificador | identificador
- SIMPLE::= LISTA_VARS = EXPRESION_LOGICA
- OPCION_DOS::= OPCION_DOS , SIMPLE | SIMPLE

- TIPO_DATO::= int || bool | float | char*
- PRINCIPAL::= int main ( ) { LISTA_SENTENCIAS } | int main ( ) { }
- METODO::= TIPO_DATO identificador (LISTA_PARAMETROS) { LISTA_SENTENCIAS } | TIPO_DATO identificador ( ) { LISTA_SENTENCIAS }
- LISTA_PARAMETROS::= LISTA_PARAMETROS , PARAMETRO | PARAMETRO
- PARAMETRO::= TIPO_DATO identificador
- LISTA_SENTENCIAS::= LISTA_SENTENCIAS SENTENCIA | SENTENCIA
- SETENCIA::= WHILE | DO_WHILE | IF_ELSE | DECLARACION | ASIGNACION_VAR | DINCREMENTOS ; | LLAMADA ; | FUNCION_IMPRIMIR | RETORNO ;
- DINCREMENTOS::= identificador ++ | identificador –
- RETORNO::= return EXPRESION_LOGICA ;
- ASIGNACION_VAR::= identificador ASIGNADORES EXPRESION_LOGICA ;
- ASIGNADORES::= -= | += | =
- FUNCION_IMPRIMIR::= print (EXPRESION_LOGICA) ;
- LLAMADA::= identificador (PARS_LLAMADA)
- PARS_LLAMADA::= PARS_LLAMADA , EXPRESION_LOGICA | EXPRESION_LOGICA
- CONDICION::= EXPRESION_LOGICA
- WHILE::= while (EXPRESION_LOGICA) { LISTA_SENTENCIAS }
- DO_WHILE::= do { LISTA_SENTENCIAS } while (EXPRESION_LOGICA) ;
- IF_ELSE::= if (EXPRESION_LOGICA) { LISTA_SENTENCIAS } ELSE | if (EXPRESION_LOGICA) { LISTA_SENTENCIAS }
- ELSE::= else { LISTA_SENTENCIAS }
- EXPRESION_LOGICA::= EXPRESION_LOGICA && EXPRESION_LOGICA | EXPRESION_LOGICA || EXPRESION_LOGICA || EXPRESION_RELACIONAL
- EXPRESION_RELACIONAL::= EXPRESION OPERADOR_RELACIONAL EXPRESION | EXPRESION
- OPERADOR_RELACIONAL::= == | != | < | > | <= | >=
- EXPRESION::= EXPRESION + EXPRESION | EXPRESION – EXPRESION | EXPRESION / EXPRESION | EXPRESION * EXPRESION | EXPRESION % EXPRESION | -EXPRESION | (EXPRESION_LOGICA) | identificador | LLAMADA | identificador ++ | identificador – | verdadero | falso | cadena | numero