**Gramática bnf**

Tarea 1

Raúl Alfaro y Diego Araya

Curso de Lenguajes de Programación

<programa> : := <decl\_glob>\* <funcion>\* <principal>

<decl\_glob> : := "let" <decl\_var> "$"

<principal> : := "procedure" "principal" "є" "э" <bloque> /\* void, sin parámetros \*/

<funcion> : := "function" <id> "є" <param\_formales\_opt> "э" ":" <tipo\_ret>

<bloque>

<param\_formales\_opt> ::= /\* vacío \*/

| <lista\_params>

<lista\_params> ::= <param> ("," <param>)\*

<param> ::= <id> ":" <tipo>

<tipo\_ret> ::= "int" | "float" | "char" | "bool"

<tipo> ::= "int"

| "float"

| "char"

| "bool"

| <tipo\_arreglo>

<tipo\_arreglo> ::= "int" "[" <tam\_arreglo> "]"

| "char" "[" <tam\_arreglo> "]"

<tam\_arreglo> ::= <lit\_entero> /\* tamaño estático, positivo \*/

/\* ===================== BLOQUES Y SENTENCIAS ===================== \*/

<bloque> ::= "¿" <sentencia>\* "?"

| <sentencia> /\* bloque reducido: una sola sentencia \*/

<sentencia> ::= <sent\_decl\_local>

| <sent\_asig\_o\_expr>

| <sent\_decide>

| <sent\_loop>

| <sent\_for>

| <sent\_return>

| <sent\_break>

| <sent\_io>

| <comentario> /\* comentarios como sentencia (se ignoran) \*/

<sent\_decl\_local> ::= "let" <decl\_var> "$"

<decl\_var> ::= <id> ":" <tipo> <init\_opt>

| <id> ":" <tipo\_arreglo> <init\_arr\_opt>

<init\_opt> ::= /\* vacío \*/

| "=" <expr>

<init\_arr\_opt> ::= /\* vacío \*/

| "=" <lit\_arreglo> /\* creación con asignación \*/

<asignacion> ::= <lvalue> "=" <expr>

<lvalue> ::= <id>

| <id> "[" <expr> "]" /\* acceso/modificación de arreglo \*/

<expr\_sin\_asignacion> ::= <expr> /\* expresiones permitidas como sentencia \*/

/\* ===================== ESTRUCTURAS DE CONTROL ====================\*/

<sent\_decide> ::= "decide" "of"

( "є" <cond\_bool> "э" "->" <bloque> )+

( "else" "->" <bloque> )?

"end" "decide" "$"