

GRAMÁTICA PROPORCIONADA EN EBNF

```
<DSL>::={<Accion>}+
<Accion>::="accion" <NombreAccion> "{" {<Comando>}+ "}"
<NombreAccion> ::= "" <Identificador> ""
<Comando> ::= "moverArchivo" "a" "" <Ruta> ""
              | "usarEscaneoProfundo"
              | "retornar" "resultado"
              | "notificar" "" <Mensaje> ""
<Identificador> ::= <Letra> {<Letra> | <Digito>}*
<Ruta> ::= {<Caracter>}+
<Mensaje> ::= {<Caracter>}+
<Letra> ::= "A" | "B" | "C" | ... | "Z" | "a" | "b" | "c" | ... | "z"
<Digito> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
<Caracter> ::= <Letra> | <Digito> | "_" | "-" | "/" | " "
```

GRAMÁTICA TRADUCIDA A ANTLR

```
grammar DSL;
programa      : accion+ EOF ;
accion       : 'accion' nombreAccion '{' comando+ '}' ;
nombreAccion  : IDENTIFICADOR ;
comando      : ('moverArchivo' 'a' ruta
               | 'usarEscaneoProfundo'
               | 'retornar' 'resultado'
               | 'notificar' mensaje
               | 'contador' '=' valor) ';' '?' ;
ruta         : CADENA ;
mensaje      : CADENA ;
valor        : ENTERO ;
IDENTIFICADOR : LETRA (LETRA | DIGITO)* ;
CADENA       : '"' (~["\r\n"])* '"' ;
ENTERO       : [0-9]+ ;

fragment LETRA : [a-zA-Z] ;
fragment DIGITO : [0-9] ;
WS           : [ \t\r\n]+ -> skip;
```