

Cómo usar: antlr lab

# antlr lab

- Es una herramienta para
  - Prototipar lenguajes
  - Verificar definiciones de lenguajes
  - Testear la estructura del árbol generado por el parser
- Elementos a definir
  - Lexer
    - Conjunto de strings que forman los tokens del lenguaje
  - Parser
    - Reglas de derivación que conforman
      - Terminales
      - No Terminales
      - Orden de presedencia

1. Lexer tab
2. Parser tab
3. Panel editable
4. Programa

1. Lexer tab

2. Parser tab

3. Panel editable

4. Programa

```
1 parser grammar ExprParser;
2 options { tokenVocab=ExprLexer; }
3
4 program
5     : stat EOF
6     | def EOF
7     ;
8
9 stat: ID '=' expr ';'
10      | expr ';'
11      ;
12
13 def : ID '(' ID (',' ID)* ')' '{' stat* '}' ;
14
15 expr: ID
16      | INT
17      | func
18      | 'not' expr
19      | expr 'and' expr
20      | expr 'or' expr
21      ;
22
23 func : ID '(' expr (',' expr)* ')' ;
```

Input: sample.expr

Start rule: program

Run

# Pasos

1. Descargue la definición provista por la materia `Lexer+Parser.pdf`
2. **Lexer**
  - a. Click en el Lexer tab (1)
  - b. Copie el texto del recuadro lexer y pegue el texto en el panel editable (3)
3. **Parser**
  - a. Click en el Parser tab (2)
  - b. Copie el texto del recuadro Parser y pegue el texto en el panel editable (3)
4. **Run**
  - a. Click en el boton (verde) Run
  - b. Si su programa esta OK, un árbol se dibujará abajo del boton
  - c. Si su programa no esta Ok, un árbol con indicaciones de error aparecerá abajo del boton.

# Programa de ejemplo I

Condicional simple ( $\text{exp1} ? \text{exp2}$ )  
Condicional completo ( $\text{exp1} ? \text{exp2} : \text{exp3}$ )  
Asignación y Condicionales

```
f(x,y) {  
    x ? x+1 ;  
    y ? y+ 1 : y-1;  
    z = y ? y+ 1 : y-1;  
}
```

# Programa de ejemplo II

Envio de mensajes sin argumento Envio de mensajes con argumento Envio de mensajes con argumentos
<pre>f(x,y) {   x.foo();   x.foo(4);   x.bar(x.foo, 4); }</pre>