

Análisis semántico



RAFAEL DUQUE MEDINA
RAFAEL.DUQUE@UCLM.ES

Índice

2

- Problema
- Análisis léxico
- Análisis sintáctico
- Análisis semántico

Problema

3

- Disponemos de un lenguaje que define la estructura de una red social
 - Los usuarios que forman parte de la red
 - Los vínculos o relaciones establecidos entre los usuarios
- El objetivo es procesar automáticamente este lenguaje y crear una interfaz que muestre gráficamente la estructura de la red social

Lenguaje de entrada e interfaz gráfica

4

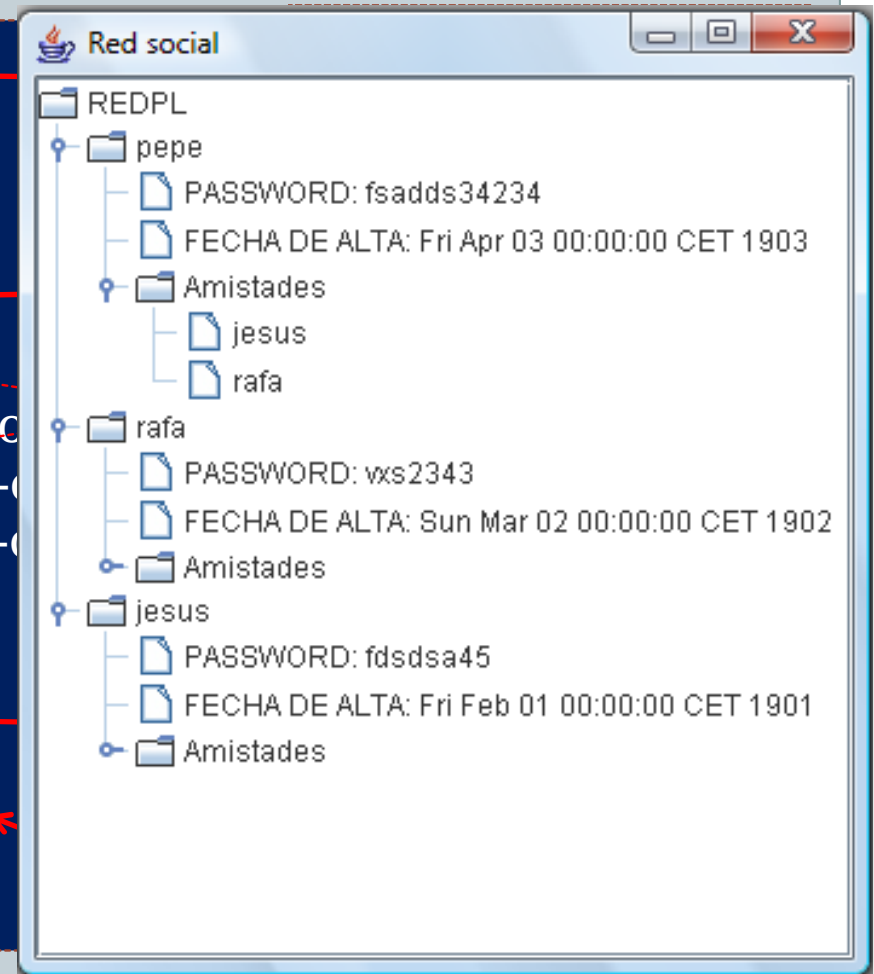
REDPL

Participantes

jesus	fdsdsa45	01-01-01
rafa	vx2343	02-02-02
pepe	ds34234	03-03-03

Vinculos

Origen: jesus	Destino: rafa;
Origen: rafa	Destino: pepe;
Origen: pepe	Destino: jesus;



Análisis léxico (.flex)

5

[A-Z]+	{return symbol(sym.NombreRed,new String(yytext()));}
[a-z]+	{return symbol(sym.Usuario,new String(yytext()));}
[a-z]+[0-9]+	{return symbol(sym.Password,new String(yytext()));}
"Participantes"	{return symbol(sym.Participantes);}
"Vinculos"	{return symbol(sym.Vinculos);}
"Origen:"	{return symbol(sym.Origen);}
"Destino:"	{return symbol(sym.Destino);}
[0-9][0-9]	{return symbol(sym.Digito,new Integer(yytext()));}
","	{return symbol(sym.PuntoyComa);}
"_"	{return symbol(sym.Guion);}

REDPL

Participantes

jesus	fdsdsa45	01-01-01;
rafa	vxs2343	02-02-02;
....		

Vinculos

Origen: jesus	Destino: rafa;
Origen: rafa	Destino: pepe;

Análisis sintáctico (.cup)

6

terminal Participantes, PuntoyComa, Vinculos, Guion, Usuario, Password, Origen, Destino, NombreRed, Digito;

non terminal P, A, R, Fecha U, V ,S;

S ::= R P U A V;

R ::= NombreRed ;

P ::= Participantes ;

U ::= Usuario Password Fecha PuntoyComa U | Usuario Password Fecha PuntoyComa ;

Fecha ::= Digito Guion Digito Guion Digito;

A ::= Vinculos;

V ::= Origen Usuario Destino Usuario PuntoyComa V | Origen Usuario Destino Usuario PuntoyComa ;

REDPL

Participantes

jesus fdsdsa45 01-01-01;

rafa vxs2343 02-02-02;

....

Vinculos

Origen: jesus Destino: rafa;

Origen: rafa Destino: pepe;

Análisis semántico

7

```
terminal PuntoyComa, Guion;  
terminal String Usuario, Password;  
terminal Integer Digito;  
non terminal Date Fecha;  
non terminal Vector<Usuario> U;
```

```
import java.util.Date ;  
class Usuario{  
    private String nombre;  
    private String password;  
    private Date fecha;  
    public Usuario (String nombre,String  
password, Date fecha){  
        this.nombre=nombre;  
        this.password=password;  
        this.fecha=fecha;  
    }  
}
```

```
U ::= Usuario:us Password:pa Fecha:fe PuntoyComa U:vector  
{:Usuario usuario=new Usuario (us,pa,fe); vector.add(usuario);RESULT=vector;:}  
|Usuario:us Password:pa Fecha:fe PuntoyComa {:Vector<Usuario> v=new Vector<Usuario>();  
Usuario usuario=new Usuario (us,pa,fe); v.add(usuario); RESULT=v;:};  
Fecha::= Digito:ad Guion Digito:bd Guion Digito:cd{: Date f=new Date  
(ad.intValue(),bd.intValue(),cd.intValue()); RESULT=f;:};
```

Análisis semántico

8

terminal PuntoyComa;
terminal **String** Origen, Destino;
non terminal **Vector<Relacion>** V;

```
class Relacion{  
  
    String origen, destino;  
  
    public Relacion(String o, String d){  
        origen=o;  
        destino=d;  
    }  
}
```

.....

V ::= Origen Usuario:**o** Destino Usuario:**d** PuntoyComa V:**vector** {:**Relacion** relacion=new **Relacion** (o,d); vector.add(relacion);**RESULT=vector;;**}|Origen Usuario:**o** Destino Usuario: **d** PuntoyComa {:**Vector<Relacion>** v=new **Vector<Relacion>**(); **Relacion** relacion=new **Relacion** (o,d); v.add(relacion); **RESULT=v;;**};

Análisis semántico

9

terminal **String** NombreRed;
non terminal P, A;
non terminal **String** R;
non terminal **Vector**<**Usuario**> U;
non terminal **Vector**<**Relacion**> V;
non terminal **String** R;

```
public class Red {  
    private String nombre;  
    private Vector<Usuario> miembros;  
    private Vector<Relacion> amistades;  
  
    public Red(String n, Vector<Usuario> m,  
        Vector<Relacion> r) {  
  
        nombre = n;  
        miembros = m;  
        amistades = r;  
    }  
}
```

S ::= R:re P U:us A V:vi{:Red red=new Red(re,us,vi); RESULT=red; :};

R ::= NombreRed:nr {:RESULT=nr;:};

Análisis semántico

terminal String NombreRed;
non terminal P, A;
non terminal String R;
non terminal Vector<Usuario> U;
non terminal Vector<Relacion> V;
non terminal String R;

```
import java.awt.*;  
import javax.swing.*;  
import javax.swing.tree.*;  
import java.util.Vector;
```

```
public class SimpleTree extends JFrame {  
    .....  
    private DefaultMutableTreeNode  
    processHierarchy(Red red) {  
        DefaultMutableTreeNode node =  
        new DefaultMutableTreeNode(red.getNombre());  
        DefaultMutableTreeNode child;  
        DefaultMutableTreeNode password;  
        DefaultMutableTreeNode fecha;  
        DefaultMutableTreeNode amigos;
```

```
.....  
S ::= R:re P U:us A V:vi{:Red red=new Red(re,us,vi); RESULT=red; SimpleTree st=new  
SimpleTree(RESULT);:};  
R ::= NombreRed:nr {:RESULT=nr;:};
```