

# Lenguajes y Autómatas I

## TAREA 26

1. En esta tarea se deberá realizar la siguiente práctica sobre el uso del JFlex.
2. En primer lugar deberá descargar el archivo que contiene todos los archivos necesarios para su uso, y que se llama *jflex-1.6.1.zip*, actualmente disponible en el sitio:  
<http://jflex.de/download.html>
3. Una vez descargado, acceda a la carpeta en la que ha sido guardado y descomprima el archivo.
4. Ahora inicie el programa *Netbeans*.
5. En el menú de herramientas (*Tools*) pase a Bibliotecas (*Libraries*).
6. Oprima el botón *New Library...* y asígnele el nombre *JFlex* y oprima *OK*.
7. Luego haga clic en *Add JAR/Folder....*
8. En la carpeta que descomprimió busque la carpeta *lib* y seleccione la librería *jflex-1.6.1.jar*
9. Haga clic en *Add JAR/Folder* para añadirla y luego oprima *OK*.
10. A continuación debe crear un nuevo proyecto (*File – New Project*) luego seleccione *Java Application*, y oprima *Next >*.
11. En *Project Name* asigne el nombre de *analizador* y oprima *Finish*.
12. Ahora tiene que crear un archivo de especificación con extensión *flex*, que contenga las reglas léxicas del analizador.
13. Para ello con el botón derecho haga clic sobre el nombre del proyecto *analizador*, seleccione *New* y luego *Other....*
14. En categorías seleccione *Other* y luego seleccione *Empty File* del lado derecho y oprima *Next>*.
15. En *File Name* escriba el nombre del archivo al que llamará *analizador.flex* y oprima *Finish*.
16. Dentro de este archivo vacío copie las reglas léxicas del analizador, el cual se divide en tres partes, cada una de ellas separada por *%%*, como se explica a continuación.
17. La primera parte serán el “*package*” que contendrá la clase generada y el “*import*”, donde se van a importar las clases que se van a utilizar:  

```
package analizador;  
import static analizador.tokens.*;  
%%
```
18. A continuación sigue la segunda parte, que se compone de las definiciones, donde se van a definir ciertos parámetros (iniciados con *%*) y los meta-símbolos deseados.  

```
%class AnalizadorLexico  
%type tokens  
Letra = [A-Za-z]  
Digito = [0-9]  
Blanco = [ \t\r\n]
```
19. Además se define el código de usuario requerido, en este caso se declara una cadena a la que se asignará el lexema que se haya encontrado.

# Lenguajes y Autómatas I

```
%{  
public String lexema;  
%}  
%%
```

20. Y la tercera parte contendrá las expresiones regulares que definen las acciones que se realizan al encontrar cada token.

```
{Blanco}+ {/* Se ignoran caracteres en blanco */}  
"//" .* {/* Se ignoran los comentarios */}  
"+" {return SUMA;}  
"- " {return RESTA;}  
"*" {return MULTIPLICA;}  
"/" {return DIVIDE;}  
"=" {return ASIGNA;}  
"==" {return IGUAL;}  
(-+)?{Digito}+ {lexema=yytext(); return NUMERO;}  
{Letra}({Letra} | {Digito})* {lexema=yytext(); return  
IDENTIFICADOR;}  
 . {return ERROR;}
```

21. Agregue una nueva clase de tipo *enum*, para ello seleccione *New File* y en la categoría *Java* seleccione *Java Enum* y oprima *Next* >.

22. Póngale el nombre de *tokens* y oprima *Finish*.

23. Aquí se van a enlistar todos los tokens, agregando la línea siguiente:

```
SUMA, RESTA, MULTIPLICA, DIVIDE, ASIGNA, IGUAL, NUMERO, IDENTIFICADOR,  
ERROR
```

24. Ahora verifique que cuenta con la biblioteca *JFlex* en la lista de *Libraries*, para ello despliegue en la lista de bibliotecas y si no aparece, oprima botón derecho y seleccione *Add Library....*

25. Localice en la lista a la biblioteca *JFlex* y oprima *Add Library*.

26. Pase ahora a *Analizador.java* para crear los métodos con los que se generará la clase de salida llamada *yylex.java*.

27. Primero, en la línea inmediata a package, indique la siguiente declaración:

```
import java.io.File;
```

28. Ahora, dentro del método principal defina un *string* llamado *path* con la trayectoria donde se localiza el archivo *analizador.flex*, usualmente está en la carpeta *NetBeansProjects*, de *Documents*, quedando algo que puede verse como sigue (observe que las diagonales no son inversas):

```
String path =  
"C:/Users/Eduardo/Documents/NetBeansProjects/analizador/src/  
analizador/analizador.flex";
```

29. A continuación agregue la llamada al método:

```
parserLexer(path) ;
```

30. Posteriormente defina el método *parserLexer* como sigue:

```
public static void parserLexer(String path) {
```

# Lenguajes y Autómatas I

```
File file = new File(path);
jflex.Main.generate(file);
}
```

31. A continuación procederá a ejecutar la clase principal (*Run File*) para poder generar la nueva clase, si todo está correcto verá que se genera la clase llamada *AnalizadorLexico.java* que será la que se utilizará en el compilador.
32. Ahora va a construir una interfaz para usar el analizador.
33. Con el botón derecho sobre el nombre del proyecto seleccione *New* y *JFrame Form....*
34. Al nuevo formulario dele el nombre de *Interface* y oprima *Finish*.
35. Agregue una etiqueta (*Label*) que diga *Ingrese el Texto* (propiedad *text*) y un cuadro de texto (*Text Field*) suficientemente grande a su lado, borre el texto en la propiedad *text*.
36. Debajo coloque un área de texto (*Text Area*) también bastante grande, donde se va a mostrar el resultado del análisis, antecedido por una etiqueta que diga *Resultado*.
37. Finalmente agregue un botón (*Button*) llamado *Analizar*, para realizar el análisis.
38. Pase a la vista de código (*Source*) de la interface y agregue el siguiente código:
39. Escriba el siguiente método le da funcionalidad al botón, haciendo la llamada al método *probar*:

```
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt) {
    try {
        probar();
    }
    catch (IOException ex) {
        System.out.println(ex.getMessage());
    }
}
```

40. A continuación transcriba el método probar:

```
public void probar() throws IOException{
    File archivo = new File("archivo.txt");
    PrintWriter Writer;
    try {
        Writer=new PrintWriter(archivo);
        Writer.print(jTextField1.getText());
        Writer.close();
    }
    catch (FileNotFoundException ex) {

        Logger.getLogger(Interface.class.getName()).log(Level.SEVERE,
        null, ex);
    }
    Reader reader = new BufferedReader(new
    FileReader("archivo.txt"));
    AnalizadorLexico lexer=new AnalizadorLexico(reader);
    String resultado="";
    while (true){
        tokens token=lexer.yylex();
```

# Lenguajes y Autómatas I

```
if (token==null) {
    resultado=resultado+"EOF";
    jTextArea1.setText(resultado);
    return;
}
switch (token){
    case ERROR:
        resultado=resultado+"Símbolo no reconocido \n";
        break;
    case IDENTIFICADOR: case NUMERO:
        resultado=resultado+"Token = "+token+"
"+lexer.lexema +" \n";
        break;
    default:
        resultado=resultado+"Token = "+token+" \n";
}
}
```

41. Se agregan las importaciones necesarias:  
**import java.io.\*;**  
**import java.util.logging.\*;**
42. Ahora ya está listo el analizador para probarlo, ejecute **Run File** y proceda a realizar el análisis de la siguiente cadena de entrada:  
**Suma := A + B \* D / 2 ^ E;**
43. Envíe la imagen de la ejecución del analizador como evidencia de la tarea.
44. Fin de la Práctica.