

INSTITUTO TECNOLÓGICO DE NUEVO LEÓN



Lenguaje y Autómatas 2

Proyecto 2

Catedrático: Juan Pablo Rosas Baldazo

Nombre: Diego Salvador Fernández García, Mauricio Tamez

Introducción

Este proyecto consiste en leer un archivo, un código. Y separa cada palabra por un signo de puntuación, cada palabra se guarda en un array para luego comparar cada token con la tabla de símbolos. Una vez que el código este haciendo su trabajo vamos a insertar un código de cualquier clase o un símbolo cualquiera así el programa podrá realizar la función de separar en token cada una de las líneas del código.

Pseudocodigo

```
public enum Tipo {//
    NUMERO ("^\\d+$"),
    OPERADOR_BINARIO ("[*|/|+|-|=]$",),
    PALABRAS_RESERVADAS ("(INICIO|FIN|
+ "ENTERO|CADENA|FLOAT) $"),
    IDENTIFICADORES("^&[A-Za-z]+$"),
    SIMBOLO ("[:]$$$");

    //SE INSERTARA UN NUMERO, OPERADOR, PALABRAS RESERVADAS ,
IDENTIFICADORES Y UN SIMBOLO PARA QUE SE PUEDA IDENTIFICAR
EN UN CODIGO A QUE FAMILIA LE PERTENECE.

    public final String patrón;

    Tipo (String s) {
        this. patron = s;
    }
}

public class Token {
    public Tipo getTipo () {
        return tipo;
    }

    //SE MUESTRA EL TIPO DE DATO

    public void setTipo (Tipo tipo) {
```

```
this. tipo = tipo;
```

```
}
```

```
//SE MODIFICA EL TIPO DE DATO
```

```
public String getValor () {
```

```
return valor;
```

```
//SE MUESTRA EL TIPO DE DATO
```

```
}
```

```
public void setValor (String valor) {
```

```
this.valor = valor;
```

```
}
```

```
//SE MODIFICA EL TIPO DE DATO
```

```
public class probar extends javax. swing.JFrame {
```

```
public probar () {
```

```
initComponents();
```

```
setLocationRelativeTo(this);
```

```
mostrar ();
```

```
while (st.hasMoreTokens()) {
```

```
//SE DIVIDE LA CADENA EN TOKENS
```

```
String palabra = st.nextToken();
```

```
//SE CREA UN ARRAY
```

```
boolean matched = false;
```

```
//SE DEVUELVE A FALSO
```

```
public void abrir() {
```

```
FileReader fr = null;
```

```
BufferedReader bf = null;
```

```
File archivo = null;
```

```
try {
```

// APERTURA DEL FICHERO Y CREACION DE BufferedReader PARA PODER HACER UNA LECTURA COMODA (DISPONER DEL METODO readLine()).

```
archivo = new File("C:\\Users\\respaldo\\Desktop\\fichero.txt");
```

```
fr = new FileReader(archivo);
```

```
bf = new BufferedReader(fr);
```

// LECTURA DEL FICHERO

```
String linea;
```

```
while ((linea = bf.readLine()) != null) {
```

```
System.out.println(linea);
```

```
txtentrada.setText(linea);
```

```
}
```

```
} catch (Exception e) {
```

```
JOptionPane.showMessageDialog(null, "Archivo no encontrado");
```

```
} finally {
```

// EN EL FINALLY CERRAMOS EL FICHERO, PARA ASEGURARNOS QUE SE CIERRA TANTO SI TODO VA BIEN COMO SI SALTA UNA EXCEPCION

```
try {
```

```
if (null! = fr) {
```

```
fr.close();
```

```
}
```

```
} catch (Exception e2) {
```

```
e2.printStackTrace();
```

```
}
```

```
}
```

```
}
```

```
public void mostrar() {
```

```

try {
dtm.addColumn("TIPO");
dtm.addColumn("TOKEN");
    dtm2.addColumn("ERRORES");

//SE MUESTRAN LOS DATOS

//METODO MAIN

// SE VERIFICA EL CODIGO Y LO SEPARA POR SUS PALABRAS
RESERVADAS

public class Lexer {

public static void main(String[] args) {
new probar().setVisible(true);
}

}

```

Conclusiones

En este proyecto se pudo conocer a las funciones y cual es su objetivo también como hacer para poder abrir archivos que no esten en java. Otro punto de dicho proyecto se destaca que se conocio las instrucciones BufferedReader y String Tokenizer la primera instrucción que funciona para poder dar una entrada y lectura de datos en un programa de java y así poder interactuar con el usuario, la segunda que funciona para poder dividir un string en substrings a base de otros String que es un separador, se logró el objetivo de poder crear tablas para comparar con cada uno de los símbolos o caracteres que se inserten adentro de un arreglo.

Bibliografía

<https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html>

<https://prezi.com/htmcq3v71qdu/unidad-ii-generacion-de-codigo-intermedio/>