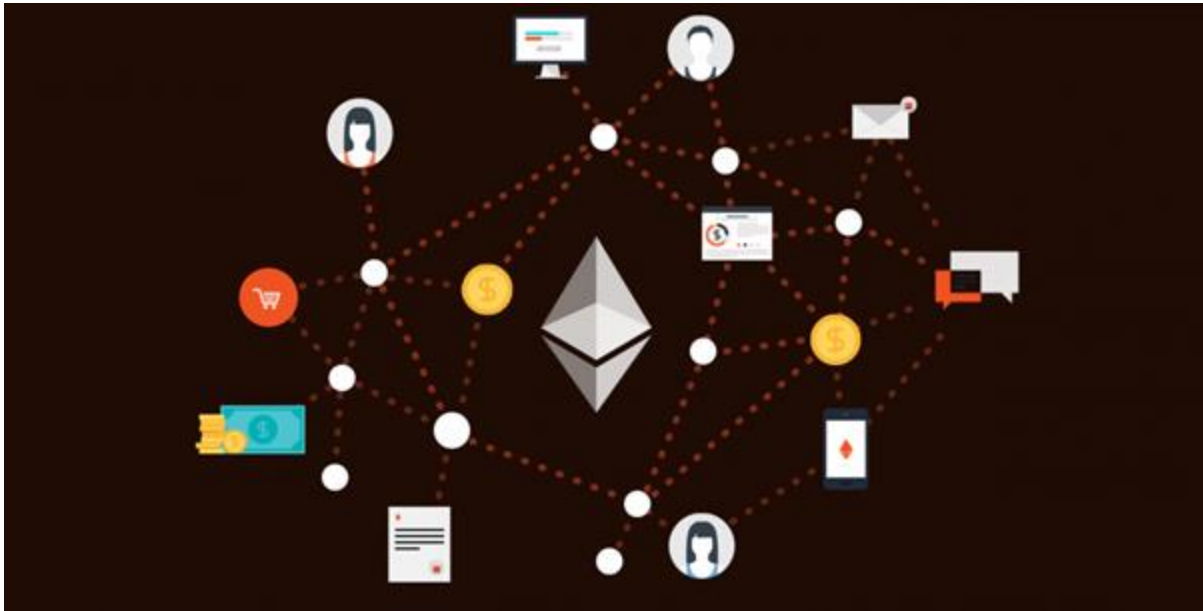




INSTITUTO TECNOLÓGICO  
DE NUEVO LEÓN



JUAN PABLO ROSAS BALDAZO

**TAREA:** PROYECTO  
UNIDAD 2

15 DE MARZO DE 2018

IVÁN ALEJANDRO PADILLA ESPARZA | #15480063  
INGENIERO EN SISTEMAS COMPUTACIONALES

# Índice

Introducción	<b>Pág. 2</b>
Pseudocódigo	<b>Pág. 3</b>
Conclusión	<b>Pág. 8</b>
Bibliografía	<b>Pág. 9</b>

# *Introducción*

La finalidad de realizar este proyecto en base a la unidad 2.

Teníamos que leer un archivo y posteriormente separar las palabras por signos especiales, dichas palabras se guardan en un array para finalmente comparar con la tabla de símbolos para comprobar a que símbolo hace referencia.

Otra de la tarea de esta unidad, era crear cuádruplas a partir de cadenas.

## Pseudocódigo

### Clase Tokenizer.java

```
clase Tokenizer {
```

```
    Método booleano isNumero (String numero) {
```

```
        Variable double num
```

```
        intenta {
```

```
            num = cambiarlo a double(el numero enviado por parámetro);
```

```
        }
```

```
        En caso de error(imprime el error){
```

```
            regresa(false);
```

```
        }
```

```
            regresa(true);
```

```
    }
```

```
    Método booleano isOperador(String operador, tabla de operadores[]){ Ciclo para  
        recorrer la tabla{
```

```
        Si(tabla operadores es igual a operador)){ regresa
```

```
            true;
```

```
        }
```

```
    }
```

```
    Si no, regresa false;
```

```
}
```

```
Método booleano isPuntuacion(String punt,tabla de signos de puntuación){
```

```
    Ciclo para recorrer la tabla {
```

```
        Si(tabla de signos de puntuación es igual a punt){ regresa  
            true;
```

```
    }
```

```
}
```

```
Si no, regresa false;
```

```
}
```

```
Método leer(String ruta){
```

```
    intenta{
```

```
        leer el archivo de la ruta;
```

```
        String linea;
```

```
        String token;
```

```
        Nuevo arreglo llamado ar;
```

```
        Mientras existan líneas en el archivo{
```

```
            Separador(línea. Que es donde leerá el separador, los símbolos
```

```
que separara: ";/././(/)/[/]/:", devuelve los símbolos separados);
```

```
            Mientras existan más tokens{
```

```
                Agrega el token al arreglo ar;
```

```
            }
```

```
        }
```

```
En caso de error(imprime el error){
```

```
}
```

```

    }

    cuadruplor (String not_pref){
        int i=0;
        char  item1, item2, operador;
        String[] operando = {"+", "-", "*", "/"};
        while (i<= not_pref.length()) {
            item1 = not_pref.charAt(i);
            for(int j=0;j<4;j++){
                if(operando[j].equals(item1)){
                    operador = item1;
                    item2 = not_pref.charAt(i+1);
                }
            }
            if(item1==operando){
                item2 = not_pref[i+i];
                if(item2==operando){
                    if(operador !=null)
                        {Agregar cuadruplo}
                }
                else{operador=item[i]}
            }
            else{operador = item[i]}
            i++;
        }
    }
}

```

```

Método principal{

    Tabla operadores = {"+", "-", "*", "/"};

    Tabla palabrasReservadas =
{"if", "while", "public", "for", "private", "main", "int", "float",
"double", "String"};

    Tabla signosDePuntuacion = {";", ":", ",", "(", ")", "[", "]", "}" }; Tabla Identificadores =
    {"01", "operador"},

    {"02", "p_reservada"},

    {"03", "numero"},

    {"04", "espacio en blanco"},

    {"05", "identificador"},

    {"06", "puntuacion"}};

    leer("C:\\Prueba\\archivo.txt");

    isPuntuacion("", signosDePuntuacion);

    isOperador("", operadores);

    isNumero("");

}

}

```

## *Conclusion*

En este reporte podemos aprender o aprendemos algo acerca de separar cadenas mediante la lectura de un archivo de texto, posteriormente hacer una tabla.

Si bien el código no funcione, dado que para nosotros venir en blanco de autómatas 1 y ahora ver esto en autómatas 2, nos tratamos de llevar el mayor conocimiento posible.



## Bibliografía

<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/colecciones/stringtokenizer.htm>

<https://www.javatpoint.com/string-tokenizer-in-java>

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

