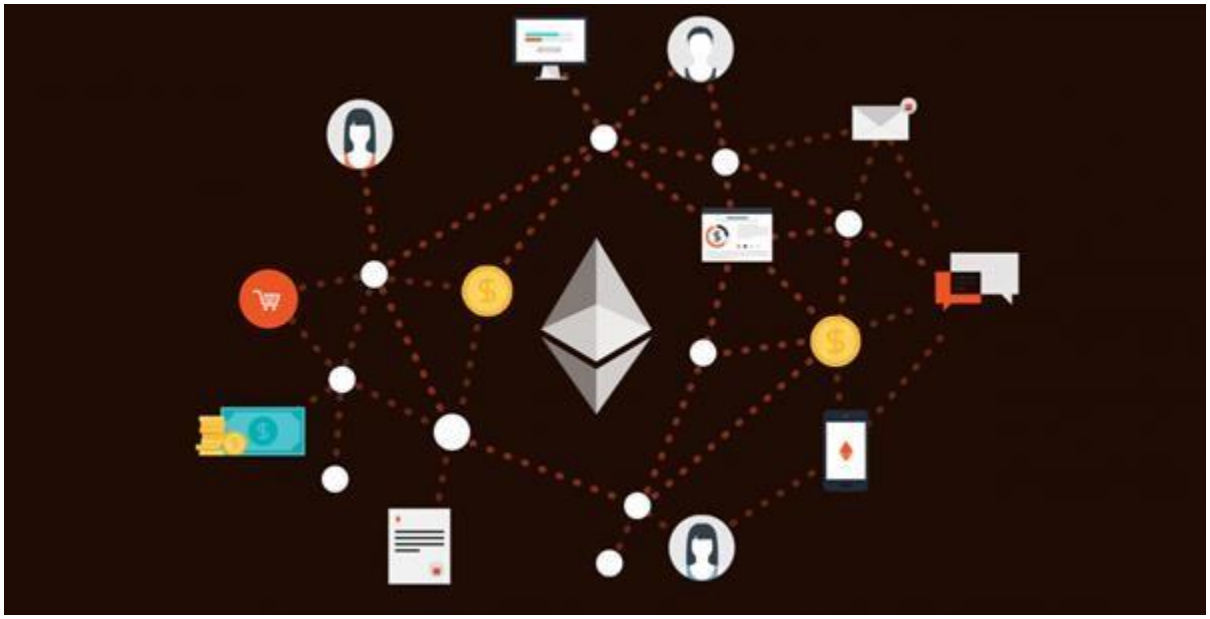
INSTITUTO TECNOLÓGICO
DE NUEVO LEÓN



JUAN PABLO ROSAS BALDAZO
TAREA: PROYECTO
UNIDAD 2

15 DE MARZO DE 2018

IVÁN ALEJANDRO PADILLA ESPARZA | #15480063
INGENIERO EN SISTEMAS COMPUTACIONALES

Introducción

En este proyecto, tenemos como objetivo realizar la lectura de un archivo externo, en esta práctica se utilizará un bloc de notas en Java. Una vez que el programa haya logrado reconocer la información que tiene el bloc de notas, lo siguiente a realizar es que, por medio del mismo código, separare carácter por carácter, esto también incluye cualquier letra, signo de puntuación, expresiones matemáticas, etc., para que al momento de ejecutar de nuevo el código, comparare la información con la tabla de símbolos, esto para conocer a que símbolo hace referencia dentro de nuestra tabla.

Una vez obtenido tu información ordenada carácter por carácter y ya comparada con la tabla de símbolos, lo que hará es crear cuádruplas ó crear una tabla con "n" filas (pero solo cuatro columnas) todo esto por medio de una cadena, para así poder acomodar los caracteres dentro de la misma cuádrupla creada.

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, regresafalse;

}

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, regresafalse;

}

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;i<4;i++){
                if(operando[i].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 proyecto aprendimos a como separar las cadenas, y así mismo a leer los archivos de texto externos. También aprendí a como crear tablas y como poder compararlas con otras tablas que nosotros mismos creamos y que una vez que se comparen, puedan agregarse dentro de un arreglo y/o un cuádruplo que nosotros mismos creamos.

Cabe recalcar que, si bien el programa no cumple con lo requerido del maestro, uno se lleva conocimiento de esto, y así poder mejorarlo en un futuro, lo rescatable en esta práctica es lo que nos llevamos, ya que veníamos de autómatas I sin conocimiento alguno.

Bibliografía

Oracle (2016) String Tokenizer, 15-Marzo-2018, de Docs Oracle Sitio Web: https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html