

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Seminario de solución de problemas de traductores de lenguajes II

“Análisis léxico”

Nombre de los alumnos:

Juan Oswaldo Cadena Perez 212652712
Alejandro Ortega Rojas 212650612
María De La Luz Ramírez Jaramillo 211733123
Samuel Julián Márquez Villarreal 213025789

Profesor: Armando Ramos Barajas
Fecha: 20 agosto de 2018
Sección: D03
Calendario: 2018 B

Índice

1.1 Introducción	2
1.2 Objetivo general	2
1.3 Objetivo particular	2
1.4 Desarrollo	3
1.5 Conclusión	11
1.6 Bibliografías	11
1.7 Apéndice	12
1.8 Acrónimos	12

1.1 Introducción

Lo que se presenta a continuación es un analizador léxico, desarrollado por nosotros desde la clasificación de las palabras que podrá utilizar el compilador, así como su descripción de cada uno de ellas, para lo que sirve así como la codificación del programa en Python, además de una demostración grafica del autómata con todas nuestras palabras reservadas y demostrar la ejecución del programa. Este actividad corresponde a la primera fase denominada Análisis léxico.

1.2 Objetivo general

Desarrollar un analizador léxico mostrando el autómata correspondiente con un mínimo de 50 palabras y mostrando su ejecución en un lenguaje de programación.

1.3 Objetivo particular

Seleccionar un conjunto de palabras para que cumplan la función de palabras reservadas dentro de nuestro compilador. Una vez establecidas nuestras palabras se procederá a realizar el autómata correspondiente que incluya todas y cada una de las palabras escogidas, terminado el autómata.

El analizador léxico acepte por lo menos las palabras más comunes en un compilador y describa su funcionamiento, el mínimo de palabras reservadas será de 50.

1.4 Desarrollo

Primero se realizó una lista de 50 palabras incluyendo los caracteres más comunes para determinar nuestro alfabeto reservado en el analizador léxico las cuales son las siguientes:

Unsigned ,unsigned int ,unsigned long ,unsigned long int ,unsigned float ,unsigned long float ,Short int ,long float ,long int ,bool ,float ,string ,char ,int ,switch ,case ,default ,break ,if ,else ,else if ,do ,while ,for ,class ,struct ,cout ,cin ,include ,catch ,try ,not ,and ,xor ,or ,return ,@ ,} ,{ ,] ,[,# ,& ,! ,) ,(,/ ,+ , - , = , < , > , * , % ,, ,: , " , ; , ,

Una vez realizado esto proseguimos a determinar nuestro autómata dentro de la plataforma JFLAP, en el cual definimos todos los estados partiendo de un estado inicial y terminando en los 50 estados finales que contiene cada una de las palabras reservadas.

A continuación proseguimos a implementar nuestro autómata dentro del lenguaje Python, primero instalamos anaconda, después creamos un txt que contiene todas las palabras reservadas que nuestro analizador léxico reconoce. Posteriormente proseguimos a realizar el código el cual consiste en un ciclo con todas las posibles palabras reservadas, en caso de que la palabra no exista esta puede ser usada como un identificador.

Este es nuestro Código en python:

```
from io import open
```

```
i = 0
```

```
with open("palabritas.txt","r") as fichero:
```

```
    for linea in fichero:
```

```
        bandera=0
```

```
        tamano = len(linea)
```

```
        linea = linea[0:tamano]
```

```
        print(linea, " tiene una cantidad de caracteres de ", len(linea))
```

```
        if("unsigned" in linea):
```

```
            print("unsigned es una palabra reservada para indicar que una variable no tiene signo")
```

```
bandera=1
if("int" in linea):
    print("palabra reservada para declarar una variable entera")
    bandera=1
if("long" in linea):
    print("palabra reservada para declarar una variable entera con un rango más amplio")
    bandera=1
if("float" in linea):
    print("palabra reservada para declarar una variable flotante real")
    bandera=1
if("short" in linea):
    print("palabra reservada para declarar una variable entera corta")
    bandera=1
if("bool" in linea):
    print("palabra reservada para declarar una variable del tipo verdadero o falso")
    bandera=1
if("string" in linea):
    print("palabra reservada para declarar una variable del tipo cadena de texto")
    bandera=1
if("char" in linea):
    print("palabra reservada para declarar una variable de un solo caracter")
    bandera=1
if("switch" in linea):
    print("palabra reservada para declarar una estructura selectiva que puede contar con casos
y un default")
    bandera=1
if("case" in linea):
    print("palabra reservada para declarar un caso dentro de la estructura selectiva switch")
    bandera=1
```

```
if("default" in linea):  
    print("palabra reservada para indicar un caso por defecto dentro de la estructura selectiva switch")  
    bandera=1  
if("break" in linea):  
    print("palabra reservada para indicar el rompimiento de una iteración o de un ciclo")  
    bandera=1  
if("if" in linea):  
    print("palabra reservada de tipo de estructura selectiva donde se debe cumplir una condicion")  
    bandera=1  
if("else" in linea):  
    print("palabra reservada de tipo de estructura selectiva, debe de ir después de un if")  
    bandera=1  
if("do" in linea):  
    print("palabra reservada para indicar el comienzo de una estructura iterativa")  
    bandera=1  
if("while" in linea):  
    print("palabra reservada del tipo de estructura iterativa que se ejecuta mientras que se cumpla una condición")  
    bandera=1  
if("for" in linea):  
    print("palabra reservada del tipo de estructura iterativa que ejecuta un numero finito de iteraciones")  
    bandera=1  
if("void" in linea):  
    print("palabra reservada para indicar que una funcion no devuelve un valor")  
    bandera=1  
if("class" in linea):  
    print("palabra reservada para indicar una estructura del tipo clase en programación orientada a objetos")
```

```
bandera=1
if("struct" in linea):
    print("palabra reservada para indicar una estructura con un conjunto de variables dentro")
    bandera=1
if("cout" in linea):
    print("palabra reservada encargada de mostrar una cadena en la consola")
    bandera=1
if("cin" in linea):
    print("palabra reservada encargada de recibir un dato por el usuario dentro de la consola")
    bandera=1
if("include" in linea):
    print("palabra reservada para indicar una librería a incluir")
    bandera=1
if("catch" in linea):
    print("palabra reservada para evitar que un error detenga la ejecución del programa")
    bandera=1
if("try" in linea):
    print("palabra reservada para indicar que el programa pruebe una serie de instrucciones a
ejecutar")
    bandera=1
if("not" in linea):
    print("palabra reservada para realizar una negación")
    bandera=1
if("and" in linea):
    print("palabra reservada para indicar una inclusión")
    bandera=1
if("xor" in linea):
    print("palabra reservada para indicar la ejecución del algoritmo de xor")
    bandera=1
```

```
if("or" in linea):
    print("palabra reservada para indicar una opcion alternativa a evaluar")
    bandera=1
if("return" in linea):
    print("palabra reservada para indicar el retorno de un parametro")
    bandera=1
if("@" in linea):
    print("Caracter aceptado por el compilador")
    bandera=1
if("}" in linea):
    print("Caracter aceptado por el compilador Delimita el final de un bloque de codigo")
    bandera=1
if("{ " in linea):
    print("Caracter aceptado por el compilador Delimita el inicio de un bloque de codigo")
    bandera=1
if("]" in linea):
    print("Caracter aceptado por el compilador se utiliza para delimitar el final de un arreglo")
    bandera=1
if("[ " in linea):
    print("Caracter aceptado por el compilador se utiliza para delimitar la apertura de un arreglo")
    bandera=1
if("#" in linea):
    print("Caracter aceptado por el compilador que se pone antes de la palabra reservada INCLUDE")
    bandera=1
if("&" in linea):
    print("Caracter aceptado por el compilador sirve para el AND logico (condicional)")
    bandera=1
if("!" in linea):
```

```
print("Caracter aceptado por el compilador sirve para la negacion")

bandera=1

if("") in linea:

    print("Caracter aceptado por el compilador sirve para llamar una funcion")

    bandera=1

if("(" in linea):

    print("Caracter aceptado por el compilador sirve para llamar una funcion")

    bandera=1

if("/") in linea:

    print("Caracter aceptado por el compilador sirve para division entre dos variables o si es
doble para comentarios")

    bandera=1

if"+" in linea):

    print("Caracter aceptado por el compilador sirve para suma entre dos variables")

    bandera=1

if("-" in linea):

    print("Caracter aceptado por el compilador sirve para resta entre dos variables")

    bandera=1

if("=" in linea):

    print("Caracter aceptado por el compilador sirve para asignar valos a una variable o si es
doble sive como condicional")

    bandera=1

if("<" in linea):

    print("Caracter aceptado por el compilador comparacion si una variable es menor que
otra")

    bandera=1

if(">" in linea):

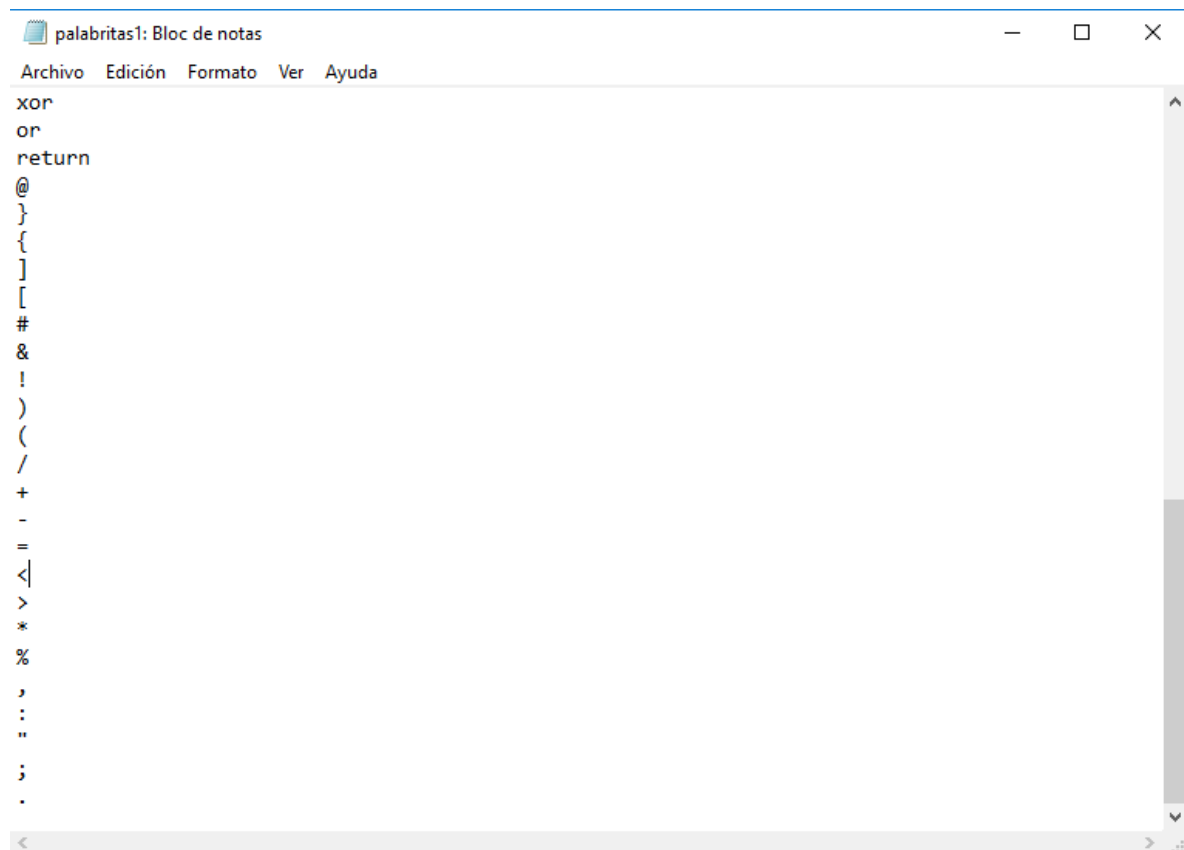
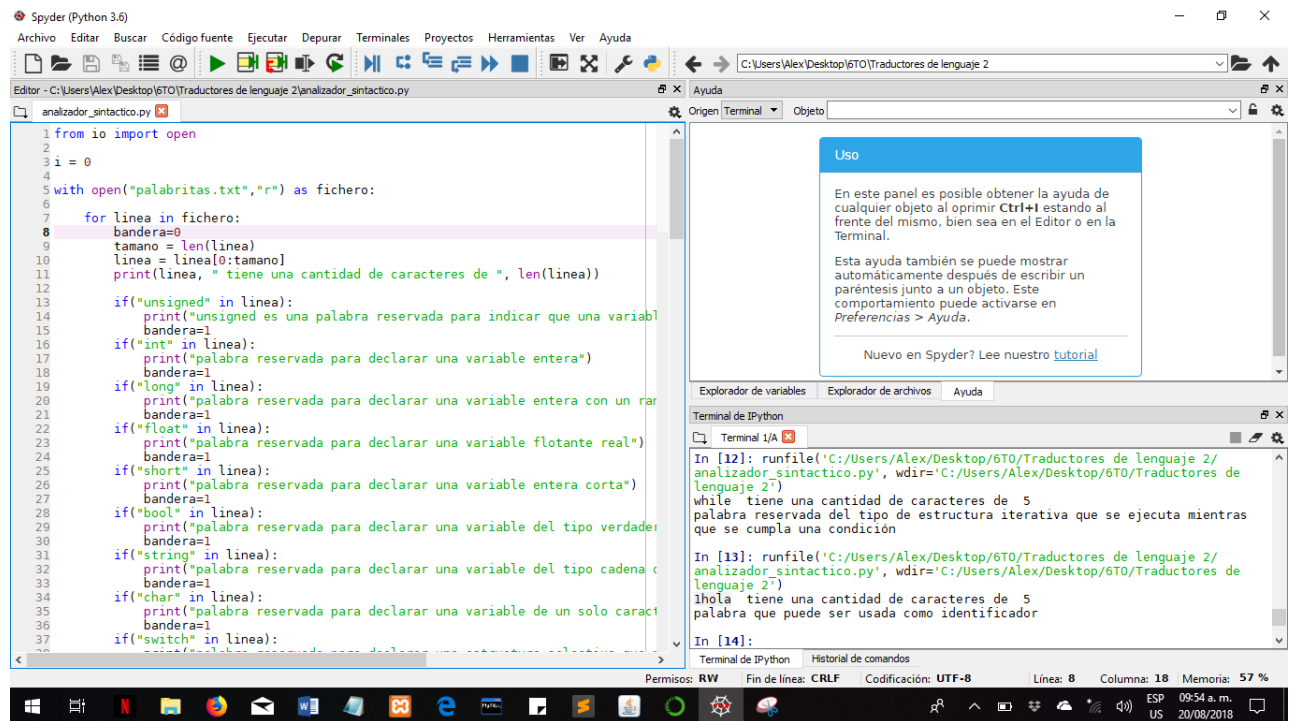
    print("Caracter aceptado por el compilador comparacion si una variable es menor que
otra")

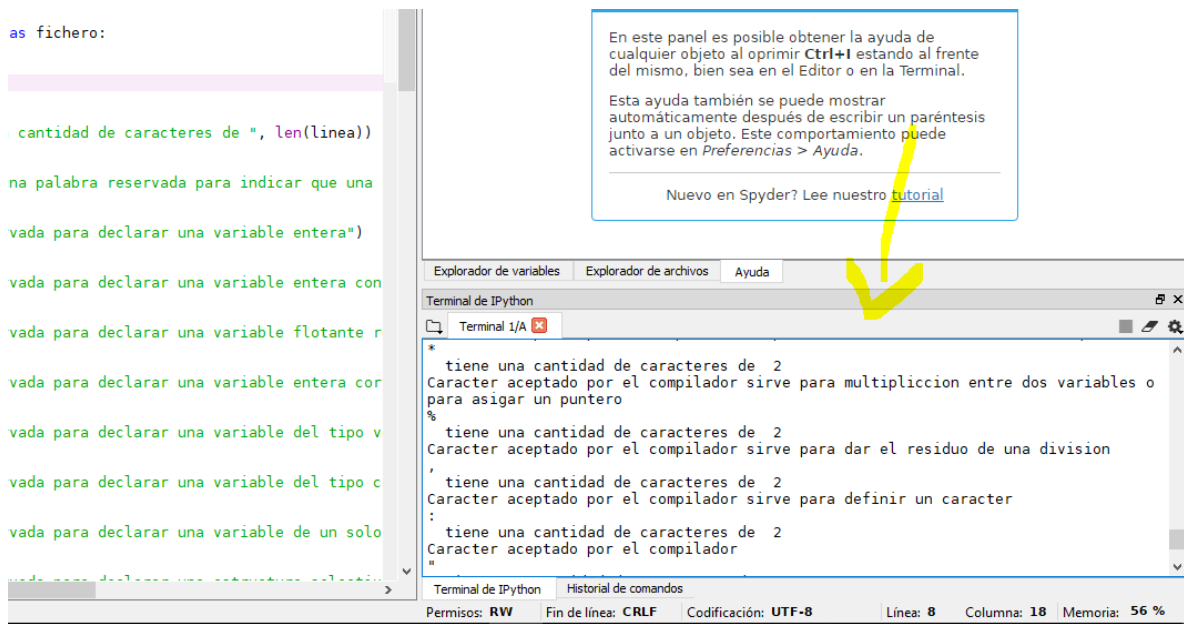
    bandera=1
```



```
if("*" in linea):  
    print("Caracter aceptado por el compilador sirve para multiplicacion entre dos variables o  
para asigar un puntero")  
    bandera=1  
if("%" in linea):  
    print("Caracter aceptado por el compilador sirve para dar el residuo de una division")  
    bandera=1  
if(", " in linea):  
    print("Caracter aceptado por el compilador sirve para definir un caracter")  
    bandera=1  
if(":" in linea):  
    print("Caracter aceptado por el compilador")  
    bandera=1  
if("'" in linea):  
    print("Caracter aceptado por el compilador")  
    bandera=1  
if('.') in linea):  
    print("Caracter aceptado por el compilador")  
    bandera=1  
if('; ' in linea):  
    print("Caracter aceptado por el compilador terminar una linea de codigo")  
    bandera=1  
  
i+=1  
if bandera == 0:  
    print("palabra que puede ser usada como identificador")  
  
fichero.close()
```

del(fichero)





1.5 Conclusión

Esta actividad nos ayudó a comprender los fundamentos de un compilador, partiendo desde un analizador léxico, fue una actividad sencilla y bastante interesante de realizar, debido a que nos puso a pensar en todas las palabras reservadas que se pueden utilizar, y además esta actividad nos ayudó a comprender de mejor forma como es que funciona el compilador, estamos ansiosos por seguir trabajando en nuestro compilador y esperamos iniciar con la siguiente fase.

1.6 Bibliografías

Anaconda

Jflap

Curso de udemy python

"unsigned"

"int"

"long"

"float"

"short"

"bool"

"string"

"char"

"switch"

"palabra reservada para declarar una estructura selectiva que puede contar con casos y un lt"

"case"
"palabra reservada para declarar un caso dentro de la estructura selectiva switch"

"default"
"palabra reservada para indicar un caso por defecto dentro de la estructura selectiva switch"

"break"
"palabra reservada para indicar el rompimiento de una iteración o de un ciclo"

"if"
"palabra reservada de tipo de estructura selectiva donde se debe cumplir una condicion"

"else"
"palabra reservada de tipo de estructura selectiva, debe de ir después de un if"

"do"
"palabra reservada para indicar el comienzo de una estructura iterativa"

"while"
"palabra reservada del tipo de estructura iterativa que se ejecuta mientras que se cumpla una condición"

"for"
"palabra reservada del tipo de estructura iterativa que ejecuta un numero finito de iteraciones"

"void"
"palabra reservada para indicar que una funcion no devuelve un valor"

"class"
"palabra reservada para indicar una estructura del tipo clase en programación orientada a objetos"

"struct"
"palabra reservada para indicar una estructura con un conjunto de variables dentro"

"cout"
"palabra reservada encargada de mostrar una cadena en la consola"

"cin"
"palabra reservada encargada de recibir un dato por el usuario dentro de la consola"

"include"
"palabra reservada para indicar una librería a incluir"

"catch"
"palabra reservada para evitar que un error detenga la ejecución del programa"

"try"
"palabra reservada para indicar que el programa pruebe una serie de instrucciones a ejecutar"

"not"
"palabra reservada para realizar una negación"

"and"
"palabra reservada para indicar una inclusión"

"xor"
"palabra reservada para indicar la ejecución del algoritmo de xor"

"or"
"palabra reservada para indicar una opcion alternativa a evaluar"

"return"
"palabra reservada para indicar el retorno de un parametro"

"@"

"Caracter aceptado por el compilador"

"}"

"Caracter aceptado por el compilador Delimita el final de un bloque de codigo"

"{"

"Caracter aceptado por el compilador Delimita el inicio de un bloque de codigo"

"]"

"Caracter aceptado por el compilador se utiliza para delimitar el final de un arreglo"

"["

"Caracter aceptado por el compilador se utiliza para delimitar la apertura de un arreglo"

"#"

"Caracter aceptado por el compilador que se pone antes de la palabra reservada INCLUDE"

"&"

"Caracter aceptado por el compilador sirve para el AND logico (condicional)"

"!"

"Caracter aceptado por el compilador sirve para la negacion"

)"

"Caracter aceptado por el compilador sirve para llamar una funcion"

"("

"Caracter aceptado por el compilador sirve para llamar una funcion"

"/"

"Caracter aceptado por el compilador sirve para division entre dos variables o si es doble para comentarios"

"+"

"Caracter aceptado por el compilador sirve para suma entre dos variables"

"_"

"Caracter aceptado por el compilador sirve para resta entre dos variables"

"="

"Caracter aceptado por el compilador sirve para asignar valores a una variable o si es doble sirve como condicional"

"<"

"Caracter aceptado por el compilador comparacion si una variable es menor que otra"

">"

"Caracter aceptado por el compilador comparacion si una variable es menor que otra"

"*"

"Caracter aceptado por el compilador sirve para multiplicacion entre dos variables o para asignar un puntero"

"%"

"Caracter aceptado por el compilador sirve para dar el residuo de una division"

" "

"Caracter aceptado por el compilador sirve para definir un caracter"

":"

"Caracter aceptado por el compilador"

""

"Caracter aceptado por el compilador"

','

"Caracter aceptado por el compilador"

','

"Caracter aceptado por el compilador terminar una linea de código"