



# TECNOLOGICO NACIONAL DE MEXICO INSTITUTO TECNOLOGICO DE CIUDAD MADERO

Alumno: Zarate Saldaña Francisco Javier.

N°Control:20070617.

Ingeniero: Lenguaje y Autómatas 1.

Fecha de entrega: 05/06/2024

Docente: Fernando Manzanares González.

6:00 pm a 7:00 pm

# Documentación Técnica: Analizadores Léxico y Sintáctico

---

## 1. Introducción

Este manual describe la estructura y funcionamiento de dos programas: un analizador léxico y un analizador sintáctico. Estos programas están implementados en Python y se utilizan para el análisis de lenguajes.

## 2. Estructura del Proyecto

El proyecto está dividido en dos carpetas principales: 'Lexico' y 'sintactico'.

## 3. Archivos Principales

### 3.1. Analizador Léxico (Lexico)

- main.py: Este archivo contiene el código principal para el analizador léxico. Define las reglas léxicas y utiliza PLY (Python Lex-Yacc) para realizar el análisis léxico.

### 3.2. Analizador Sintáctico (sintactico)

- sintactico.py: Este archivo contiene el código principal para el analizador sintáctico. Define la gramática y utiliza PLY para realizar el análisis sintáctico.

## 4. Detalles de Implementación

### 4.1. Analizador Léxico (main.py)

El archivo main.py define las expresiones regulares para los tokens y las acciones a realizar cuando se reconocen estos tokens.

```
import ply.lex as lex
```

```
# Lista de tokens
```

```
tokens = (  
    'IDENTIFIER',  
    'NUMBER',  
    # otros tokens...  
)
```

```
# Reglas de expresiones regulares para tokens simples
```

```

t_IDENTIFIER = r'[a-zA-Z_][a-zA-Z_0-9]*'
t_NUMBER = r'\d+'

# Reglas de expresiones regulares con acciones
def t_newline(t):
    r'\n+'
    t.lexer.lineno += len(t.value)

# Manejo de errores
def t_error(t):
    print(f'Caracter ilegal '{t.value[0]}')
    t.lexer.skip(1)

# Construir el lexer
lexer = lex.lex()

```

## 4.2. Analizador Sintáctico (sintactico.py)

El archivo sintactico.py define la gramática del lenguaje y las acciones semánticas para cada regla de producción.

```

import ply.yacc as yacc

# Importar tokens desde el analizador léxico
from lexico import tokens

# Reglas de la gramática
def p_statement_expr(p):
    'statement : expression'
    print(p[1])

def p_expression_binop(p):
    '''expression : expression '+' expression
                  | expression '-' expression'''
    p[0] = ('binop', p[2], p[1], p[3])

# Manejo de errores
def p_error(p):
    print(f'Error de sintaxis en '{p.value}')

# Construir el parser
parser = yacc.yacc()

```

## 5. Ejecución del Programa

Para ejecutar el analizador léxico, utilice el comando:

```
python Lexico/main.py
```

Para ejecutar el analizador sintáctico, utilice el comando:

```
python sintactico/sintactico.py
```

## 6. Manejo de Errores

Ambos programas incluyen manejo de errores para caracteres ilegales y errores de sintaxis. Estos errores se imprimen en la consola con un mensaje descriptivo.

## 7. Conclusión

Esta documentación proporciona una visión general de la estructura y funcionamiento de los programas léxico y sintáctico. Para más detalles, revise los comentarios y el código fuente en los archivos proporcionados.