



# Tecnológico de Monterrey

**Implementación de métodos computacionales (Gpo 602)**

## **Evidencia 1 Resaltador de sintaxis**

Amilka Daniela Lopez Aguilar - A01029277

María Rivera Gutiérrez - A01029678

Mauricio Emilio Monroy González - A01029647

Campus Santa Fe

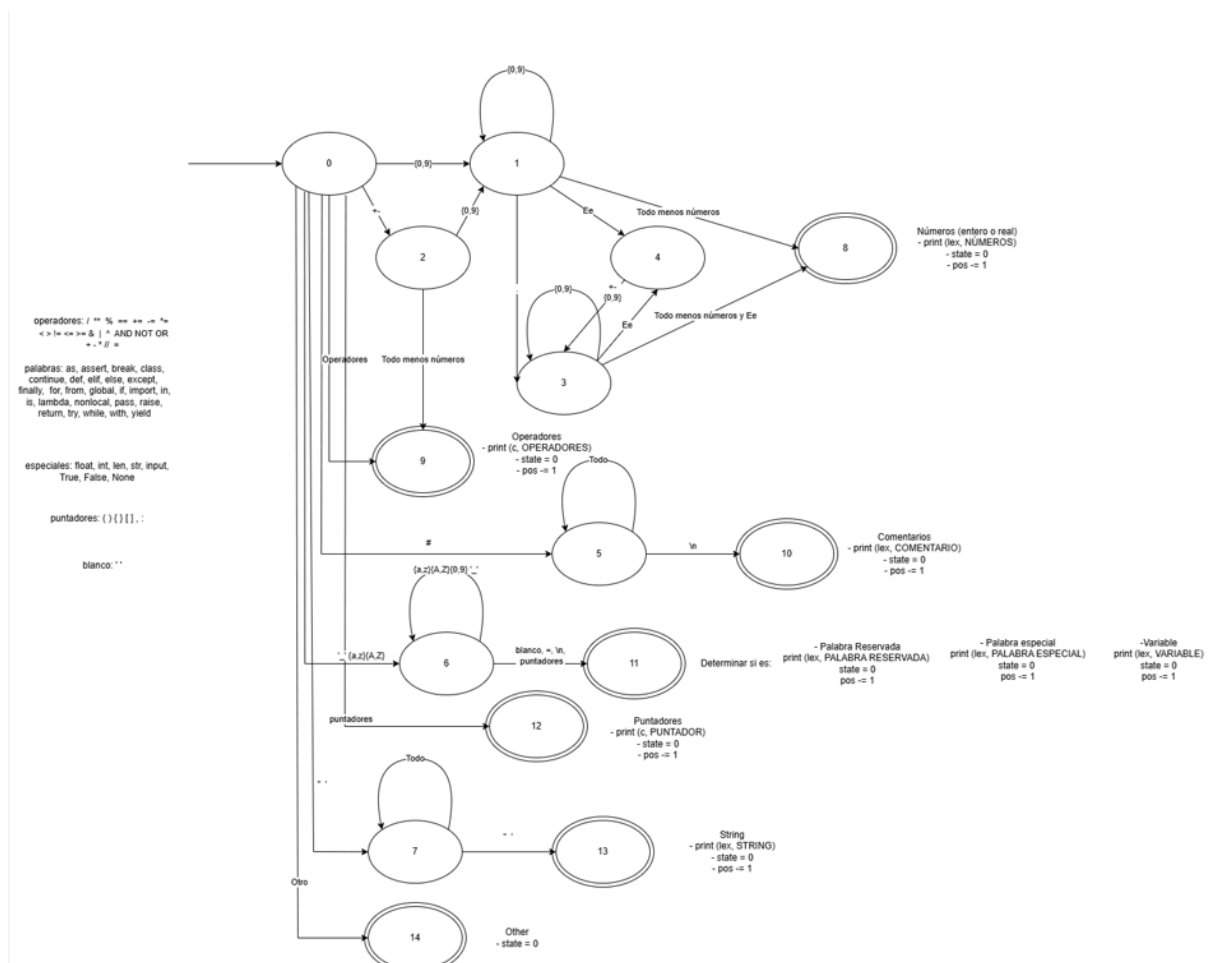
Victor de la Cueva

10 de abril de 2025

## Evidencia 1 Resaltador de sintaxis

### Desarrollo de DFA y programación

Para la elaboración de un lexer que fungirá como base para un resaltador de sintaxis de un lenguaje Python, en una página básica de HTML, fue necesario construir un grafo representando un Autómata Finito Determinístico, o DFA por sus siglas en inglés. A continuación, se muestra el diagrama de transición correspondiente a los estados determinados para las posibles entradas subsecuentes al leer la cadena de texto del programa.



Partiendo del DFA, se generó la siguiente tabla de transición para representar y condensar la función de transición correspondiente a cada estado y entrada, al igual que su correspondiente salida. Es decir, formalmente definiendo el DFA:

- $Q = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
- $\Sigma$ :
  - Símbolos para dígitos del 0 al 9
  - Operadores:  $/ * \% = + - < > ! \& | ^$
  - Puntadores:  $()\{\}\[, :$
  - Letras: símbolos para alfabeto latino
  - $. \text{“ ‘ \#}$
  - Blanco:  $\backslash n \backslash b \backslash t$
- $q_0 = 0, q_0 \in Q$
- $F = \{8, 9, 10, 11, 12, 13, 14\}, F \subseteq Q$
- $\delta$  (función que mapea  $Q \times \Sigma \rightarrow Q$ :  $\delta(q, a)$ ) representada por la tabla de transición:

Estado	Números	Operadores	Puntadores	Letras	.	Ee	\n	=	"'	#	+	-	blanco	Otro
0	1	9	12	6	14	6	15	9	7	5	2	2	15	14
1	1	8	8	8	3	4	8	8	8	8	8	8	8	14
2	1	9	9	9	9	9	9	9	9	9	9	9	9	14
3	3	8	8	8	14	4	8	8	8	8	8	8	8	14
4	3	3	14	14	14	14	14	14	14	3	3	14	14	14
5	5	5	5	5	5	10	5	5	5	5	5	5	14	14
6	6	14	11	6	14	6	11	11	14	14	14	14	11	14
7	7	7	7	7	7	14	7	13	7	7	7	7	14	14
15	15	15	15	15	15	15	15	15	15	15	15	15	15	14

Tomando eso como punto de partida, se programó un script en Python para recibir un archivo .py cualquiera, leerlo y categorizar sus contenidos para ser resaltados con colores correspondientes a sus asignaciones. A continuación, se muestra la tabla de token y tipo asignado para la prueba según el DFA utilizado en el programa. La tabla permite identificar fallos en la identificación de los tokens, inconsistencias y que la asignación entre token y tipo sea correcta. Aquellos tokens tipo TOKEN, se comportan de la misma manera, pero al momento de asignarles un color, se verifica a qué diccionario pertenece.

## Salida Obtenida en consola del programa

```
|-----|
|   Token   |   Tipo   |
|-----|
|#Test 1    |  COMENTARIOS |
|   i       |   TOKEN      |
|=          | OPERADORES   |
|   1       |  NUMEROS     |
| while     |   TOKEN      |
|   i       |   TOKEN      |
|<          | OPERADORES   |
|   10      |  NUMEROS     |
|:          | PUNTADORES   |
| for       |   TOKEN      |
|   j       |   TOKEN      |
| in        |   TOKEN      |
| range     |   TOKEN      |
| (         | PUNTADORES   |
|   5       |  NUMEROS     |
| )         | PUNTADORES   |
|:         | PUNTADORES   |
| print     |   TOKEN      |
| (         | PUNTADORES   |
| "Hello World " | STRING      |
| )         | PUNTADORES   |
| print     |   TOKEN      |
| (         | PUNTADORES   |
|   i       |   TOKEN      |
| )         | PUNTADORES   |
| print     |   TOKEN      |
| (         | PUNTADORES   |
|   j       |   TOKEN      |
| )         | PUNTADORES   |
|   i       |   TOKEN      |
| +         | OPERADORES   |
| =         | OPERADORES   |
|   1       |  NUMEROS     |
|#Test2     |  COMENTARIOS |
|# Seccion de Variables |  COMENTARIOS |
| edad      |   TOKEN      |
| =         | OPERADORES   |
|   0       |  NUMEROS     |
| year      |   TOKEN      |
| =         | OPERADORES   |
|   0       |  NUMEROS     |
| estim     |   TOKEN      |
| =         | OPERADORES   |
|   0       |  NUMEROS     |
|#Test3     |  COMENTARIOS |
| edad      |   TOKEN      |
| =         | OPERADORES   |
| int       |   TOKEN      |
| (         | PUNTADORES   |
| input     |   TOKEN      |
| (         | PUNTADORES   |
| "Edad de usuario " | STRING      |
| )         | PUNTADORES   |
| )         | PUNTADORES   |
```

year		TOKEN	
=		OPERADORES	
int		TOKEN	
(		PUNTADORES	
input		TOKEN	
(		PUNTADORES	
"Current year"		STRING	
)		PUNTADORES	
)		PUNTADORES	
estim		TOKEN	
=		OPERADORES	
year		TOKEN	
-		OPERADORES	
edad		TOKEN	
+		OPERADORES	
100		NUMEROS	
print		TOKEN	
(		PUNTADORES	
"The year in which the user turns 100 is "		STRING	
,		PUNTADORES	
estim		TOKEN	
)		PUNTADORES	

El programa pide hacer la impresión del programa fuente y la asignación de los colores en un documento HTML. A continuación, se muestra la asignación de los colores según el tipo de token reconocido.

## HTML

```
#Test 1
i = 1
while i < 10:
    for j in range(5):
        print("Hello World")
        print(i)
        print(j)
    i += 1

#Test2
# Seccion de Variables
'''Hello world'''
edad = 0
year = 0
estim = 0
Salu2_Error











#Test3
#Codigo comienza aqui
edad = int(input("Edad de usuario:"))
year = int(input("Current year:"))
estim = year - edad + 100
print("The year in which the user turns 100 is:", estim)

#Test4
# Operaciones algebraicas
a = 5 + 3.2 # Suma de entero y real
b = 10 - 4 # Resta de enteros
c = 2.5 * 1.2 # Multiplicacion de reales
d = 100 / 4 # Division de enteros
e = 1.2e3 + 3.4e-2 # Suma con notacion cientifica
f = (a + b) * c - d / e # Operacion combinada

print("Resultado de a:", a)
print("Resultado de b:", b)
print("Resultado de c:", c)
print("Resultado de d:", d)
print("Resultado de e:", e)
print("Resultado de f:", f)
```

## Comportamiento de programa

### Colores utilizados

```
.keyword { color: goldenrod; font-weight: bold; }
.special { color: lightpink; }
.variable { color: lightblue; }
.operator { color: white; }
.number { color: lightgreen; }
.comment { color: green; font-style: italic; }
.string { color: lightcoral; }
.punctuation { color: yellow; }
.error { text-decoration: underline wavy red; color: red; }
```

### Notas

- keyword se refiere al diccionario que contiene un arreglo llamado palabras
- special se refiere al diccionario que contiene un arreglo llamado especiales
- variable es un diccionario que contiene los caracteres mayúscula y minúscula del alfabeto, así como un \_ underscore
- operator es un diccionario que contiene los operadores, por ejemplo, +, \*, =, /
- number se refiere al diccionario nums
- comment se refiere al # y la evaluación de todo lo que venga después del mismo
- string se refiere al “ “ ‘ ‘ y la evaluación de todo lo que viene después.
- punctuation se refiere al puntador.
- .error se refiere a la presencia de cualquiera de los tokens no bien escritos o al estado otro.