

Universidad Rafael Landívar

Facultad de Ingeniería

Ingeniería en Informática y Sistemas

Lenguajes formales y autómatas – Sección 1

Ing. Julio David Requena Duarte



## **MANUAL DE USUARIO**

Adonis Anthony David Hernández Pérez

Carné: 1086223

Guatemala, 7 de noviembre de 2025

## Descripción del programa

Este programa simula una Máquina de Turing (MT) sencilla y evalúa cinco expresiones regulares del enunciado. Trabaja con un archivo de entrada llamado cadenas.txt (una cadena por línea). Para cada cadena puedes ver la animación del proceso (cinta, cabezal y estado) y comprobar por regex si la cadena cumple la expresión (coincidencia total). También puedes procesar todas las cadenas y guardar un reporte en resultados.txt.

### Interfaz:

- Botón Cargar cadenas.txt.
- Lista de 5 expresiones del enunciado.
- Botones Anterior / Siguiente / Reiniciar, Paso y Automático.
- Botón “Simular TODAS (5×N)” que genera resultados.txt y en la ventana muestra: Se generó 'resultados.txt' (5×N).

## Instrucciones de instalación y ejecución

### 1. Obtener el código desde GitHub

- Opción A: Clonar: usa `git clone <URL-de-tu-repositorio>` y entra a la carpeta del proyecto.
- Opción B: ZIP: descarga el ZIP desde GitHub (Code → Download ZIP) y descomprímelo.

### 2. Requisitos

- Python 3.x instalado. No se requieren librerías externas adicionales.

### 3. Ejecutar

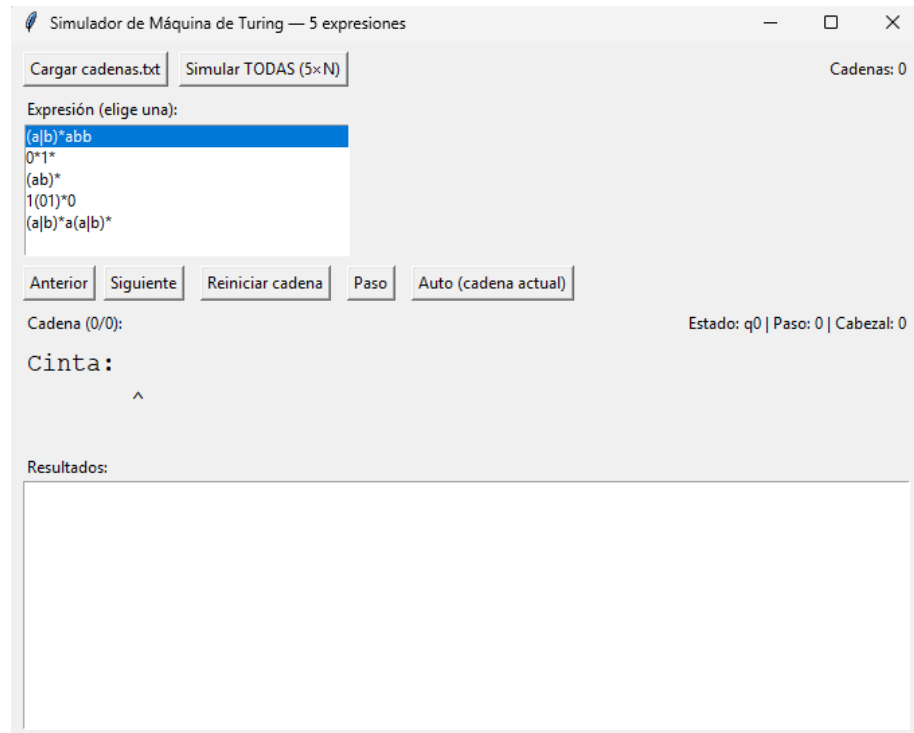
- En Visual Studio Code: abre la carpeta, abre `index.py` y ejecuta (F5) o en la terminal integrada `python index.py`.
- Desde terminal: ubicándote en la carpeta del proyecto, ejecuta `python index.py`.

### 4. Uso básico

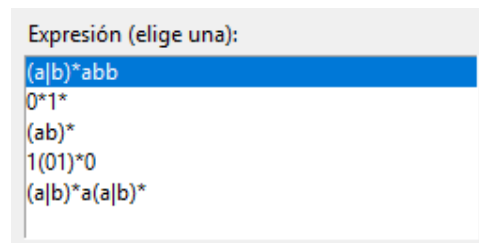
- Carga `cadenas.txt` usando el botón.
- Elige una de las 5 expresiones.
- Navega con Anterior / Siguiente o Reiniciar cadena.
- Usa Paso para ir paso a paso o Automático para ejecutar toda la cadena.
- Usa Simular TODAS (5×N) para procesar todas las cadenas con las 5 expresiones. Se crea el archivo `resultados.txt` para el detalle.

## Capturas de pantalla

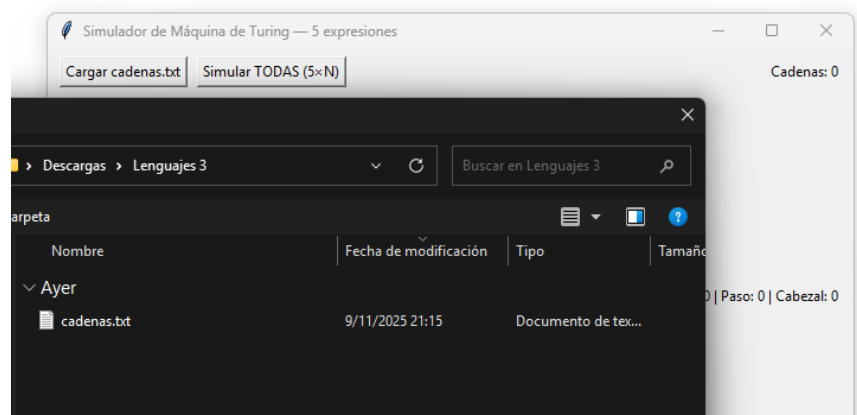
- **Interfaz inicial:**



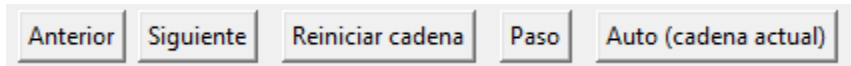
- **Expresiones:**



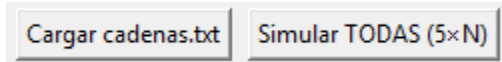
- **Carga de archivo “cadenas.txt”:**



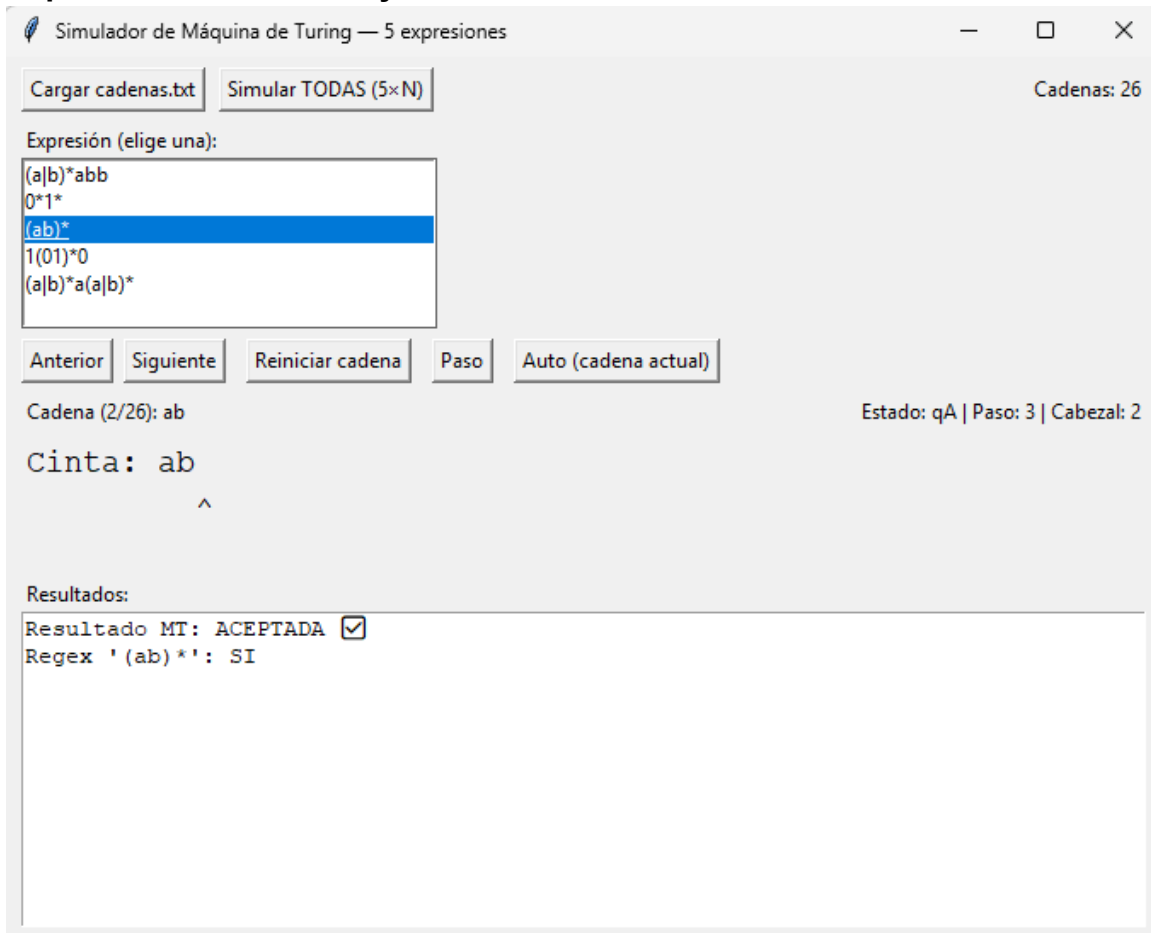
- **Botones de control y análisis de cadenas:**



- **Botones para ingreso de cadenas y creación de registro:**



- **Expresión seleccionada y cadena analizada:**



- Archivo “resultados.txt” de cadenas evaluadas con las expresiones:

```

resultados.txt
1 [(a|b)*abb] '' => MT:RECHAZADA | REGEX:NO
2 [(a|b)*abb] 'ab' => MT:RECHAZADA | REGEX:NO
3 [(a|b)*abb] 'abab' => MT:RECHAZADA | REGEX:NO
4 [(a|b)*abb] 'aba' => MT:RECHAZADA | REGEX:NO
5 [(a|b)*abb] 'a' => MT:RECHAZADA | REGEX:NO
6 [(a|b)*abb] 'abb' => MT:ACEPTADA | REGEX:SI
7 [(a|b)*abb] 'aabb' => MT:ACEPTADA | REGEX:SI
8 [(a|b)*abb] 'ababb' => MT:ACEPTADA | REGEX:SI
9 [(a|b)*abb] 'abba' => MT:RECHAZADA | REGEX:NO
10 [(a|b)*abb] 'ab' => MT:RECHAZADA | REGEX:NO
11 [(a|b)*abb] '0' => MT:RECHAZADA | REGEX:NO
12 [0*1*] '' => MT:ACEPTADA | REGEX:SI
13 [0*1*] 'ab' => MT:RECHAZADA | REGEX:NO
14 [0*1*] 'abab' => MT:RECHAZADA | REGEX:NO
15 [0*1*] 'aba' => MT:RECHAZADA | REGEX:NO
16 [0*1*] 'a' => MT:RECHAZADA | REGEX:NO
17 [0*1*] 'abb' => MT:RECHAZADA | REGEX:NO
18 [0*1*] 'aabb' => MT:RECHAZADA | REGEX:NO
19 [0*1*] 'ababb' => MT:RECHAZADA | REGEX:NO
20 [0*1*] 'abba' => MT:RECHAZADA | REGEX:NO
21 [0*1*] 'ab' => MT:RECHAZADA | REGEX:NO
22 [0*1*] '0' => MT:ACEPTADA | REGEX:SI
23 [(ab)*] '' => MT:ACEPTADA | REGEX:SI
24 [(ab)*] 'ab' => MT:ACEPTADA | REGEX:SI
25 [(ab)*] 'abab' => MT:ACEPTADA | REGEX:SI
26 [(ab)*] 'aba' => MT:RECHAZADA | REGEX:NO
27 [(ab)*] 'a' => MT:RECHAZADA | REGEX:NO
28 [(ab)*] 'abb' => MT:RECHAZADA | REGEX:NO
29 [(ab)*] 'aabb' => MT:RECHAZADA | REGEX:NO
30 [(ab)*] 'ababb' => MT:RECHAZADA | REGEX:NO
31 [(ab)*] 'abba' => MT:RECHAZADA | REGEX:NO
32 [(ab)*] 'ab' => MT:ACEPTADA | REGEX:SI
33 [(ab)*] '0' => MT:RECHAZADA | REGEX:NO
34 [1(01)*0] '' => MT:RECHAZADA | REGEX:NO
35 [1(01)*0] 'ab' => MT:RECHAZADA | REGEX:NO
36 [1(01)*0] 'abab' => MT:RECHAZADA | REGEX:NO
37 [1(01)*0] 'aba' => MT:RECHAZADA | REGEX:NO
38 [1(01)*0] 'a' => MT:RECHAZADA | REGEX:NO
39 [1(01)*0] 'abb' => MT:RECHAZADA | REGEX:NO
40 [1(01)*0] 'aabb' => MT:RECHAZADA | REGEX:NO
41 [1(01)*0] 'ababb' => MT:RECHAZADA | REGEX:NO
42 [1(01)*0] 'abba' => MT:RECHAZADA | REGEX:NO
43 [1(01)*0] 'ab' => MT:RECHAZADA | REGEX:NO
44 [1(01)*0] '0' => MT:RECHAZADA | REGEX:NO
45 [(a|b)*a(a|b)*] '' => MT:RECHAZADA | REGEX:NO
46 [(a|b)*a(a|b)*] 'ab' => MT:ACEPTADA | REGEX:SI
47 [(a|b)*a(a|b)*] 'abab' => MT:ACEPTADA | REGEX:SI
48 [(a|b)*a(a|b)*] 'aba' => MT:ACEPTADA | REGEX:SI
49 [(a|b)*a(a|b)*] 'a' => MT:ACEPTADA | REGEX:SI
50 [(a|b)*a(a|b)*] 'abb' => MT:ACEPTADA | REGEX:SI
51 [(a|b)*a(a|b)*] 'aabb' => MT:ACEPTADA | REGEX:SI
52 [(a|b)*a(a|b)*] 'babb' => MT:ACEPTADA | REGEX:SI
53 [(a|b)*a(a|b)*] 'ababb' => MT:ACEPTADA | REGEX:SI
54 [(a|b)*a(a|b)*] 'abba' => MT:ACEPTADA | REGEX:SI
55 [(a|b)*a(a|b)*] 'ab' => MT:ACEPTADA | REGEX:SI
56 [(a|b)*a(a|b)*] '0' => MT:RECHAZADA | REGEX:NO

```

## **Explicación de las expresiones regulares utilizadas**

**$(a|b)^*abb$ : termina en 'abb'**

- Acepta:
  - abb
  - aabb
  - babb
  - abababb
- No acepta:
  - ab
  - abba
  - aab

**$0^*1^*$ : ceros seguidos de unos**

- Acepta:
  - 
  - 0
  - 00
  - 1
  - 11
  - 00111
- No acepta:
  - 101
  - 010
  - 11001

**$(ab)^*$ : repeticiones del bloque 'ab' (incluye vacío)**

- Acepta:
  - 
  - ab
  - abab
  - ababab

- No acepta:

- a
- b
- aba
- abb

**$1(01)^*0$ : inicia con 1, repeticiones de 01, termina con 0**

- Acepta:

- 10
- 1010
- 101010

- No acepta:

- 0
- 1
- 100
- 1011

**$(a|b)^*a(a|b)^*$ : contiene al menos una 'a'**

- Acepta:

- a
- ba
- ab
- babab
- aa

- No acepta:

- 
- b
- bbb

**Repositorio GitHub:** <https://github.com/AnthonyH-1/Maquina-de-Turing.git>