

# **INSTITUTO TECNOLÓGICO DE ENSENADA**

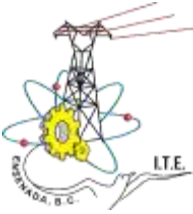
**Materia: Lenguajes y Autómatas**

**Trabajo: TAREA 01 - Autómatas Finitos  
deterministas**

**Alumna: Ariam Anette Zurita Torres**

**Docente: Xenia Padilla**

**Fecha: 17/Octubre/2025**



## 🦊 El Perrito Guardián 🦊

Diseña un AFD que acepte todas las cadenas sobre  $\{a,b\}$  que contienen la subcadena "aba".

"Como el perrito que guardaba la casa con amor, este autómata busca 'aba' con fervor"

Para la primera actividad, se nos solicitó diseñar un Autómata Finito Determinista (AFD) capaz de reconocer un lenguaje específico sobre el alfabeto  $\Sigma = \{a, b\}$ .

### 1. Definición del Lenguaje

El autómata debe aceptar todas las cadenas que contengan la subcadena "aba"

$\Sigma = \{a, b\}$

$W =$  La cadena  $w$  debe contener la subcadena "aba".

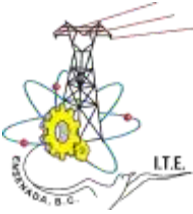
$L = \{ \text{"aba", "abab", "baba", "aabaa", "abababa", ...} \}$

### 2. Diseño y Componentes Formales

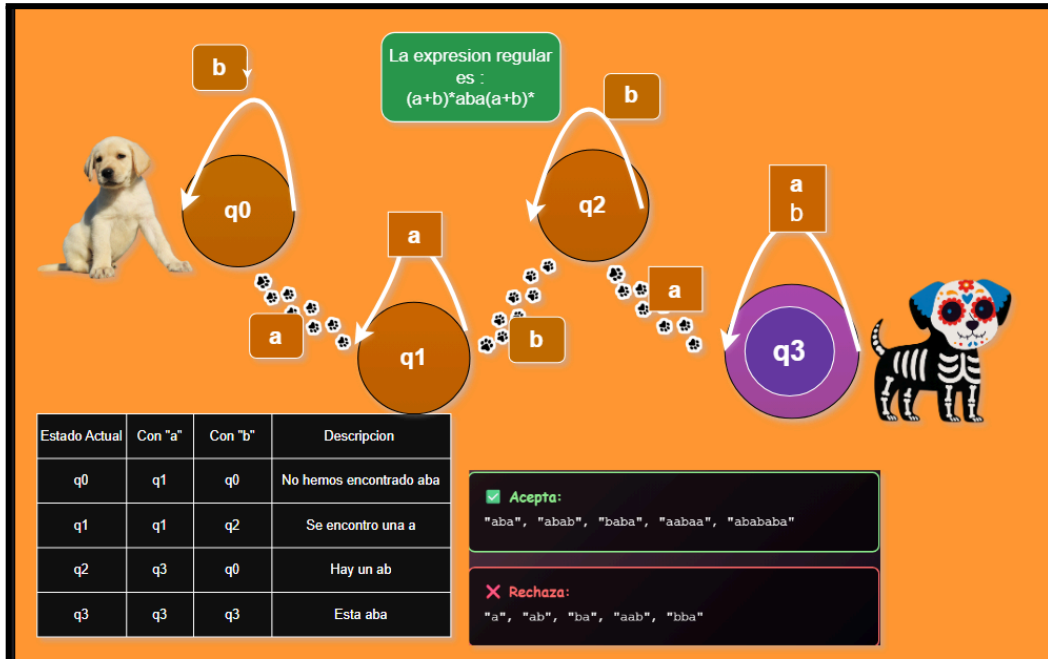
Para el diseño del autómata se utilizó la herramienta en línea [Draw.io](https://draw.io), permitiendo una representación visual clara de los estados y sus transiciones. La lógica del autómata se basa en "recordar" qué parte de la subcadena "aba" se ha encontrado.

- $q_0$ : Estado inicial. No se ha encontrado ninguna parte de "aba".
- $q_1$ : Se ha encontrado una 'a'.
- $q_2$ : Se ha encontrado la secuencia "ab".
- $q_3$ : Se ha encontrado la secuencia completa "aba". Es un estado de aceptación y "sin retorno"; una vez que se encuentra la subcadena, la cadena es aceptada sin importar los símbolos siguientes.

La expresión regular que define este lenguaje es:  $(a|b)^*aba(a|b)^*$ .

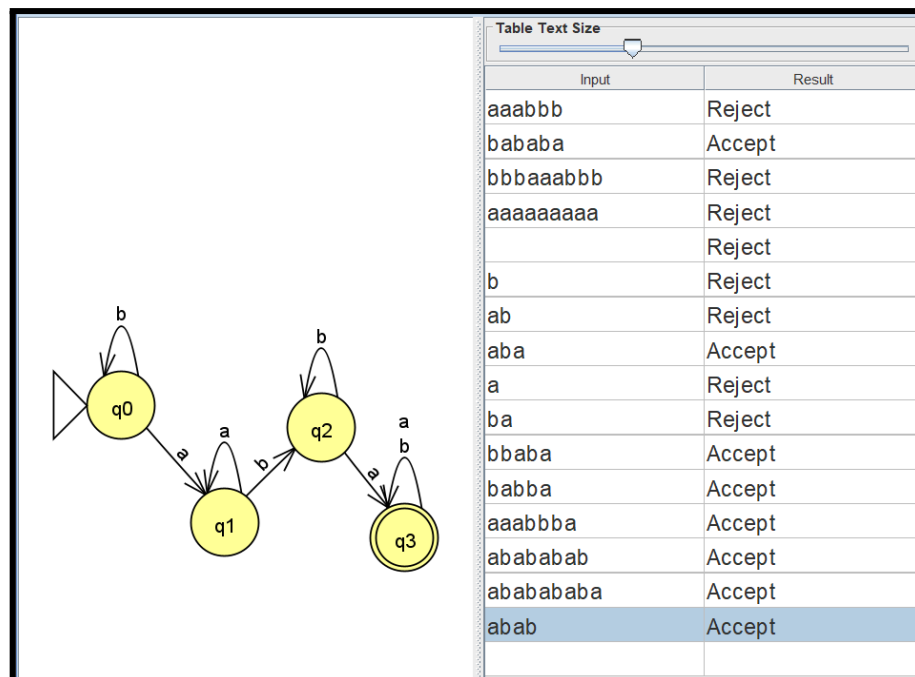


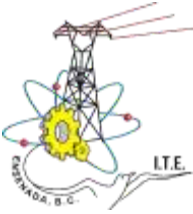
A continuacion la representacion en [Draw.io](https://draw.io)



### 3. Comprobación y Resultados

Para verificar el correcto funcionamiento del diseño, se uso el software JFLAP. Como se muestra en la siguiente imagen, el diagrama fue probado con un conjunto de cadenas de aceptación y rechazo. Los resultados confirman que el autómata reconoce el lenguaje de manera exitosa, validando que el diseño es correcto.





### ❀ El Gatito de los Tres Pasos ❀

Crea un AFD que acepte cadenas sobre  $\{a,b\}$  cuya longitud sea múltiplo de 3.

"Como el gatito que daba tres pasos en su caminar, este autómata cuenta de tres en tres sin vacilar"

Para la segunda actividad, se diseñó un AFD para reconocer cadenas cuya longitud es un múltiplo de 3.

#### 1. Definición del Lenguaje

El autómata debe aceptar cadenas sobre el alfabeto  $\Sigma = \{a, b\}$  si su longitud total es un múltiplo de 3

$\Sigma = \{a, b\}$

$W$  = Debe ser un múltiplo de 3.

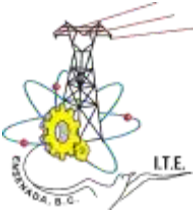
$L = \{\epsilon, "aaa", "abb", "bbbbaaa"\}$ .

#### 2. Diseño y Componentes Formales

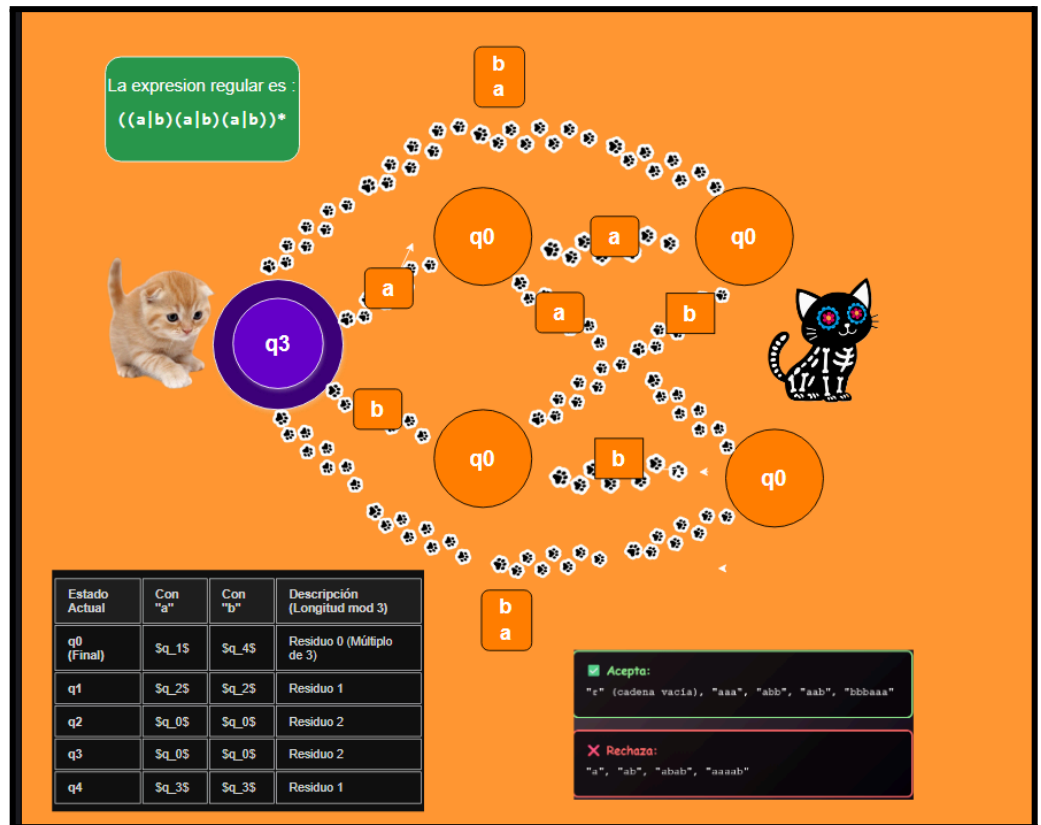
El diseño, ilustrado en [Draw.io](https://draw.io) y probado en JFLAP, utiliza una configuración de cinco estados para implementar un contador cíclico (módulo 3). Este diseño es funcionalmente correcto. Algunos estados son equivalentes y cumplen la misma función en el conteo.

- $q_0$ : Estado inicial y final. Representa una longitud de 0, 3, 6, etc. (residuo 0).
- $q_1, q_4$ : Estados que representan una longitud con residuo 1.
- $q_2, q_3$ : Estados que representan una longitud con residuo 2.

La expresión regular que define este lenguaje es:  $((a|b)(a|b)(a|b))^*$ .

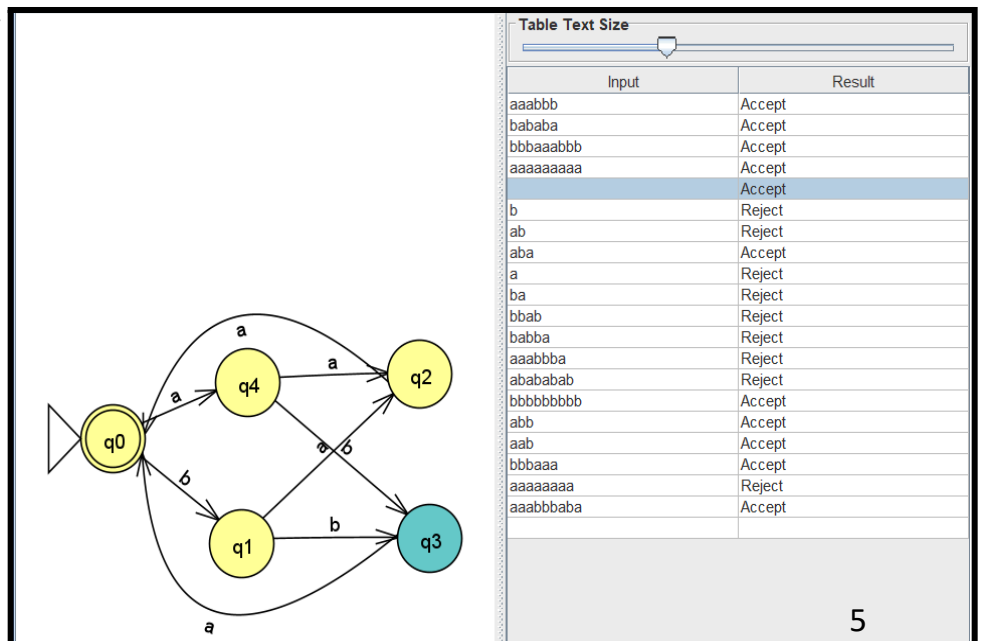


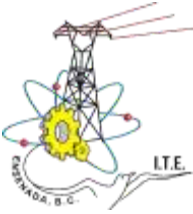
Aquí se muestra el diseño del diagrama hecho en [Draw.io](https://draw.io)



### 3. Comprobación y Resultados

El autómata de 5 estados se implementó y probó en JFLAP. Las pruebas confirmaron que el diseño reconoce correctamente todas las cadenas cuya longitud es múltiplo de 3 y rechaza todas las demás, validando esta solución específica.





## 🌿 El Loro que Contaba 'a's Pares 🌿

Diseña un AFD que acepte cadenas sobre  $\{a,b\}$  que contengan un número par de símbolos 'a' (incluyendo cero 'a's).

"Como el loro que repetía palabras de dos en dos, este autómatas cuenta las 'a's con voz"

Para la tercera actividad, se diseñó un AFD para reconocer cadenas con un número par de símbolos 'a'.

### 1. Definición del Lenguaje

El autómatas debe aceptar cadenas sobre el alfabeto  $\Sigma = \{a, b\}$  si contienen un número par de 'a's (0, 2, 4, etc.).

$$\Sigma = \{a, b\}$$

$W$  = El número de 'a's en la cadena  $w$  es par.

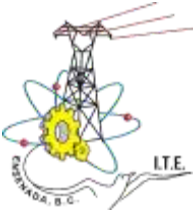
$$L = \{\epsilon, "b", "aa", "bbaa", "aabb"\}$$

### 2. Diseño y Componentes Formales

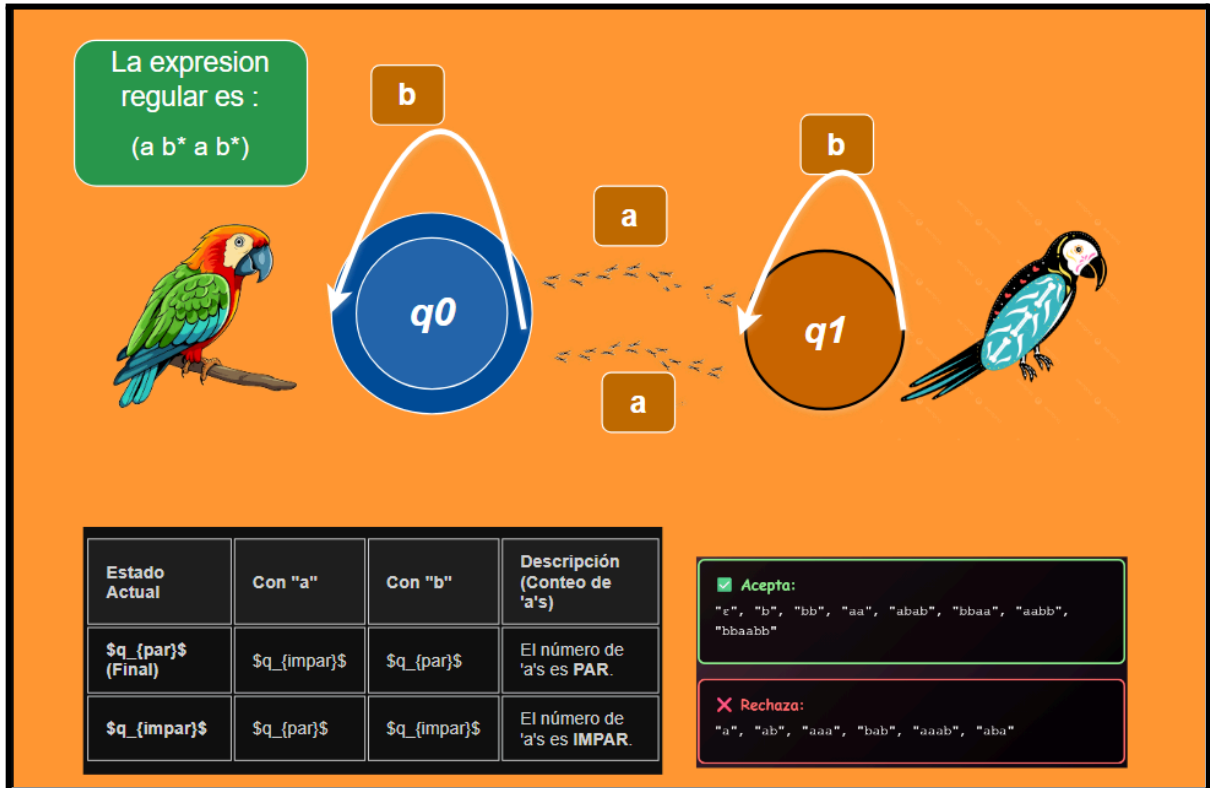
El diseño utiliza dos estados para llevar un registro de la paridad del conteo de 'a's. El símbolo 'b' no afecta el conteo y, por lo tanto, no provoca un cambio de estado.

- $q_0$  Estado inicial y final. El conteo de 'a's es par (cero al inicio).
- $q_1$ : El conteo de 'a's es impar.

La expresión regular que define este lenguaje es:  $b^* (a b^* a b^*)^*$ .



Aquí se muestra el diagrama hecho con [Draw.io](https://draw.io)

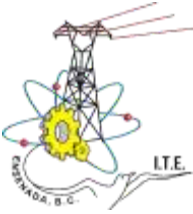


### 3.. Comprobación y Resultados

La implementación en JFLAP y las pruebas subsiguientes demostraron que el autómata acepta correctamente todas las cadenas con un número par de 'a's y rechaza aquellas con un número impar, confirmando la validez del diseño.

Table Text Size	
Input	Result
	Accept
b	Accept
bb	Accept
aa	Accept
abab	Accept
bbaa	Accept
aabb	Accept
bbaabb	Accept
a	Reject
ab	Reject
aaa	Reject
bab	Reject
aaab	Reject
aba	Accept

7



## 🥕 El Conejo que Termina en 'bb' 🥕

Crea un AFD que acepte cadenas sobre  $\{a,b\}$  que terminen con la subcadena "bb".

"Como el conejo que dejaba dos huellas al final, este autómata busca 'bb' terminal"

Para la cuarta actividad, se diseñó un AFD para reconocer cadenas que terminan con la subcadena "bb".

### 1. Definición del Lenguaje

El autómata debe aceptar cadenas sobre el alfabeto  $\Sigma = \{a, b\}$  si sus dos últimos símbolos son "bb".

$\Sigma = \{a, b\}$

$W$  = La cadena  $w$  termina con "bb".

$L = \{"bb", "abb", "aabb", "babb"\}$

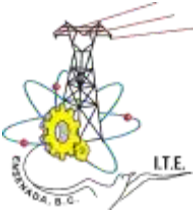
### 2. Diseño y Componentes Formales

El diseño requiere tres estados para recordar si el final de la cadena coincide con el patrón "bb".

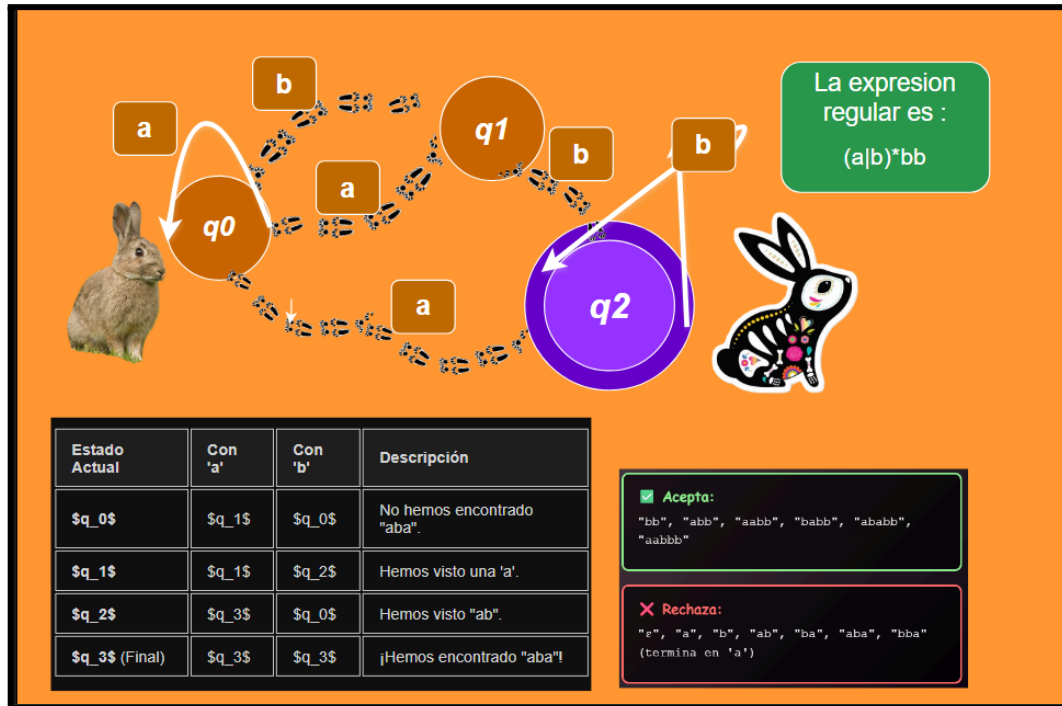
- $q_0$ : Estado inicial. La cadena no termina en 'b'.
- $q_1$ : La cadena termina con una 'b'.
- $q_2$ : Estado final. La cadena termina con "bb".

La expresión regular que define este lenguaje es:  $(a|b)^*bb$ .



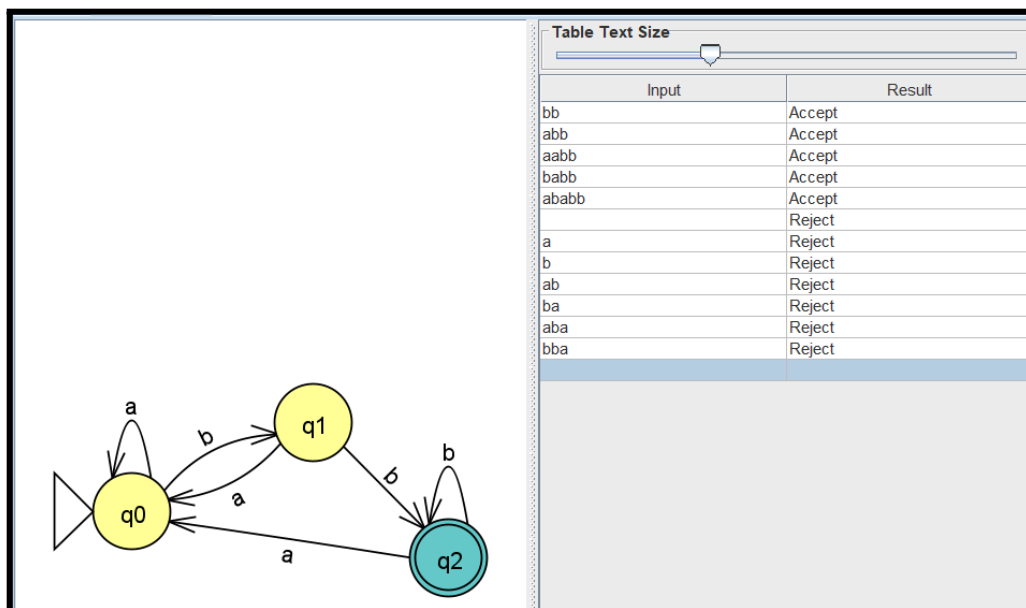


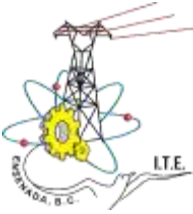
Tenemos la ultima ilustracion hecha en [Draw.io](https://draw.io) para el diagrama del Automata



### 3. Comprobación y Resultados

El modelo fue verificado en JFLAP. Las pruebas con diversas cadenas confirmaron que el autómata acepta únicamente aquellas que terminan con la subcadena "bb", validando el diseño propuesto.





## Conclusion

A lo largo de este proyecto, se ha demostrado con éxito el diseño y la aplicación de Autómatas Finitos Deterministas (AFD) para resolver problemas específicos de reconocimiento de patrones en lenguajes formales. Mediante el uso de herramientas como Draw.io para el diseño visual y JFLAP para la comprobación funcional, se validó la correcta implementación de cada uno de los cuatro autómatas propuestos.

El proceso de construcción, prueba y depuración de cada AFD no solo reforzó la comprensión teórica de conceptos como estados, transiciones y estados de aceptación, sino que también subrayó la importancia de la lógica precisa en la resolución de problemas computacionales. La temática del "Altar de las Mascotas Eternos" sirvió como un recordatorio de que incluso los conceptos más abstractos de la ciencia de la computación pueden ser abordados con creatividad y un toque humano. En definitiva, este trabajo representa una aplicación práctica y exitosa de la teoría de autómatas, demostrando su capacidad para modelar reglas y reconocer lenguajes de manera eficiente y predecible.