

Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), septiembre-octubre 2025,
Volumen 9, Número 5.

https://doi.org/10.37811/cl_rcm.v9i5

APLICACIÓN GRÁFICA PARA EL LENGUAJE FORMAL DE UNA MÁQUINA DE TURING ESTÁNDAR

**GRAPHICAL APLICATION FOR FORMAL LANGUAGE OF A
STANDAR TURING MACHINE**

Ofelia Gutiérrez Giraldi

Tecnológico Nacional de México, México

Lizbeth Sánchez Ferrer

Tecnológico Nacional de México, México

Martha Martínez Moreno

Tecnológico Nacional de México, México

Honorio Acosta Ruiz

Tecnológico Nacional de México, México

Javier Alberto Palacios López

Tecnológico Nacional de México, México

DOI: https://doi.org/10.37811/cl_rcm.v9i5.20688

Aplicación Gráfica para el Lenguaje Formal de una Máquina de Turing Estándar

Ofelia Gutiérrez Giraldi¹ofelia.gg@veracruz.tecnm.mx<https://orcid.org/0000-0002-9544-4784>Tecnológico Nacional de México /Instituto
Tecnológico de Veracruz
México**Lizbeth Sánchez Ferrer**lizbeth.sf@veracruz.tecnm.mxTecnológico Nacional de México /Instituto
Tecnológico de Veracruz
México**Martha Martínez Moreno**Martha.mm@toluca.tecnm.mx<https://orcid.org/0000-0003-3793-6315>Tecnológico Nacional de México /Instituto
Tecnológico de Veracruz
México**Honorio Acosta Ruiz**honorio.acosta.ruiz@outlook.com<https://orcid.org/0009-0008-0235-1230>Tecnológico Nacional de México /Instituto
Tecnológico de Veracruz
México**Javier Alberto Palacios López**javi.261280@gmail.comTecnológico Nacional de México /Instituto
Tecnológico de Veracruz
México

RESUMEN

El presente trabajo describe el desarrollo de una herramienta gráfica interactiva diseñada para representar y simular el lenguaje formal de una Máquina de Turing estándar (MT). La aplicación surge como herramienta de apoyo para los estudiantes de Ingeniería en Sistemas Computacionales en la materia *Lenguajes y Autómatas I* del Instituto Tecnológico de Veracruz, con la finalidad de facilitar la comprensión de los fundamentos matemáticos y lógicos que sustentan la construcción de compiladores, al resolver ejercicios prácticos a través de esta nutren el pensamiento lógico del estudiante, permitiendo mejorar la formación lógica-matemática.

Palabras clave: máquina de turing, simulación, herramienta didáctica

¹ Autor principal.

Correspondencia: ofelia.gg@veracruz.tecnm.mx

Graphical Application for Formal Language of a Standar Turing Machine

ABSTRACT

This paper describes the development of an interactive graphical tool designed to represent and simulate the formal language of a standard Turing Machine (TM). The application is designed as a support tool for students of Computer Systems Engineering in the subject Languages and Automata I at the Instituto Tecnológico de Veracruz, with the aim of facilitating the understanding of the mathematical and logical foundations that support the construction of compilers, by solving practical exercises through this, they nourish the student's logical thinking, allowing to improve the logical-mathematical training.

Keywords: turing machine, simulation, educational tool

Artículo recibido 22 septiembre 2025
Aceptado para publicación: 26 octubre 2025



INTRODUCCIÓN

La investigación se desarrolló en el Instituto Tecnológico de Veracruz, para apoyar a los estudiantes de Ingeniería en Sistemas Computacionales, que cursan la materia de Lenguajes y Autómatas I, mediante el diseño e implementación de un algoritmo que apoye la descripción de lenguajes formales, específicamente el modelo matemático de máquina de Turing que ayuda fortalecer el desarrollo de la lógica-matemática de los estudiantes. El análisis que conlleva el resolver ejercicios a través de esta nutren el pensamiento lógico del estudiante, esta formación lógica-matemática resulta de vital importancia para el desarrollo de los compiladores, el objetivo de la materia Lenguajes y autómatas es desarrollar un compilador, de ahí la importancia del aprendizaje del modelo matemático de la máquina de Turing.

La estructura del modelo de la máquina de Turing es simple, pero se nos presenta como una herramienta para plasmar cualquier algoritmo.

Las Máquinas de Turing son el antecedente y base del funcionamiento de todas estas estructuras, en el caso específico de la materia Lenguajes y Autómatas I, impartida en el Tecnológico Nacional de México campus Veracruz este tema es el último aplicado según lo marca el plan de estudio justo por la razón que el estudiante debe razonar y relacionar todas las estructuras para poder comprender su funcionamiento lógico.

Derivado de lo anterior es importante que los estudiantes cuenten con un material didáctico que fortalezca la comprensión de los tópicos antes mencionados.

Para la implementación del software, primero se recopiló la información correspondiente, haciendo un comparativo de aquellos algoritmos encontrados para tal fin; si no se asegura la funcionalidad que se busca, se optará por el diseño paso a paso del mismo. Posteriormente, se procedió al diseño del entorno gráfico que permita la comprensión y facilita el manejo de las estructuras básicas, buscando que dicho entorno sea amigable e intuitivo para los estudiantes. Una vez logrado un producto deseable, se realizaron pruebas que permitieron encontrar errores y corregir los mismos hasta obtener esta herramienta que aporta al estudiante un apoyo en la resolución de problemas.

Objetivos

General

Desarrollar un algoritmo para la simulación de una Máquina de Turing Estándar.

Específicos

- Comparar algoritmos existentes para determinar cuál es el óptimo para su utilización en múltiples campos.
- Seleccionar, adaptar o diseñar un algoritmo que permita la simulación de una máquina de Turing Estándar.
- Identificar los pasos claves para realizar la conversión.

Marco Teórico

El desarrollo del proyecto se dividió en cuatro distintas etapas:

- Investigación teórica.
- Desarrollo de Algoritmo.
- Prueba Piloto y Entorno Gráfico.
- Resultados.

Durante la primera etapa se investigaron fundamentos teóricos, definiciones formales matemáticas de ambas estructuras y sus características más relevantes.

Conceptos fundamentales

La investigación pretendió que el estudiante diseñe una Máquina de Turing estándar y además simular su funcionamiento con ayuda de la herramienta didáctica en forma de software que les permitirá aprender mejor el tema; para definir cómo se trabajo, es necesario tener conceptos del modelo matemático de la máquina de Turing.

Lenguajes

Un lenguaje se define como un conjunto discernible de secuencias que pertenecen a un alfabeto específico. Sin embargo, esta definición resulta bastante inclusiva, ya que abarca incluso lenguajes abstractos como \emptyset (conjunto vacío) o $\{\epsilon\}$ que es un conjunto que consiste únicamente en la cadena vacía (A.V.Aho, 2008).



La única restricción importante sobre lo que puede ser un lenguaje es que todos los alfabetos son finitos. De este modo, los lenguajes, aunque pueden tener un número infinito de cadenas, están restringidos a que dichas cadenas estén formadas por los símbolos que definen un alfabeto finito y prefijado (Kelly, 2002).

Lenguajes Formales

En matemáticas, lógica y ciencias de la computación, un lenguaje formal es un conjunto finito o infinito, de cadenas definidas sobre un alfabeto finito. El nombre lenguaje se justifica porque las estructuras que con este se forman tienen reglas de sintaxis (gramática) e interpretación semántica (significado) en una forma muy similar a los lenguajes hablados o naturales. Un posible alfabeto sería, digamos, $\{a,b\}$, y una cadena cualquiera sobre este alfabeto sería, por ejemplo, “ababba”. Un lenguaje sobre este alfabeto, que incluyera esta cadena, sería por ejemplo el conjunto de todas las cadenas que contienen el mismo número de símbolos a que b . La palabra vacía (esto es, la cadena de longitud cero) se permite en este tipo de lenguajes. Para referirnos a ella usamos el símbolo « ϵ », o simplemente la llamaremos “épsilon” (TALFI, 2009).

Lenguaje Informal de Máquina de Turing Estándar (MT)

La manera más práctica de comprender el funcionamiento de un MT es a través de esta definición la cual nos plantea lo siguiente:

La máquina de Turing modela matemáticamente a una máquina que opera mecánicamente sobre una cinta. En esta cinta hay símbolos que la máquina puede leer y escribir, uno a la vez, usando un cabezal lector/escritor de cinta. La operación la determina un conjunto finito de instrucciones elementales.

Más precisamente, una máquina de Turing consta de:

- Una cinta: que se divide en celdas, una al lado de la otra. Cada celda contiene un símbolo de algún alfabeto finito. El alfabeto contiene un símbolo especial llamado blanco (aquí escrito como 'B') y uno o más símbolos adicionales. La cinta se supone que es arbitrariamente extensible hacia la izquierda y hacia la derecha, o sea, la máquina de Turing siempre se provee con tanta cinta necesaria para computar. Las celdas que no se hayan escrito previamente se asumen que están rellenas con el símbolo blanco. En algunos modelos la cinta tiene un extremo izquierdo marcado con un símbolo especial; la cinta se extiende o es indefinidamente extensible

hacia la derecha.

- Un cabezal: que puede leer y escribir símbolos en la cinta y mover la cinta a la izquierda y a la derecha una (y solo una) celda a la vez. En algunos modelos el cabezal se mueve y la cinta es estacionaria.
- Un registro de estado: que almacena el estado de la máquina de Turing, uno de los estados finitos. Hay un estado inicial especial con el que el registro de estado se inicia. Turing escribe que estos estados reemplazan el "estado de la mente" en que ordinariamente estaría una persona realizando cálculos.
- Una tabla: finita de instrucciones (llamada ocasionalmente como tabla de acción o función de transición). Las instrucciones son usualmente 5-tuplas: $q_i a_j \rightarrow q_l a_j d_k$, (a veces 4-tuplas), que, dado el estado (q_i) en que la máquina se encuentra actualmente y el símbolo (a_j) que se está leyendo en la cinta (el símbolo actualmente debajo del cabezal) le indica a la máquina hacer lo siguiente en secuencia (para los modelos de 5-tupla):
 - Borra o escribe un símbolo (reemplazando a_j con a_l), y entonces.
 - Mueve el cabezal (que es descrito por d_k y puede tener los valores: 'L' para un paso a la izquierda, o 'R' para un paso a la derecha, o 'N' para permanecer en el mismo lugar) y luego asume el mismo o un nuevo estado como prescrito (ve al estado q_l).

En los modelos de 4-tupla, son especificadas como instrucciones separadas: borrar o escribir un símbolo (a_j) y mover el cabezal a la izquierda o la derecha (d_k). Específicamente, la tabla indica a la máquina: (ia) borrar o escribir un símbolo o (ib) mover el cabezal a la izquierda o a la derecha, y luego (ii) asumir el mismo o un nuevo estado, pero no las dos acciones (ia) y (ib) en la misma instrucción. En algunos modelos, si no hay ninguna entrada en la tabla para la actual combinación de símbolo y estado, la máquina se detendrá; otros modelos requieren que estén llenas todas las entradas. (Turing, 1948).

Lenguaje Formal de una Máquina de Turing Estándar

Una máquina de Turing es un modelo computacional que realiza una lectura/escritura de manera automática sobre una entrada llamada cinta, generando una salida en esta misma.

Este modelo está formado por un alfabeto de entrada y uno de salida, un símbolo especial llamado



blanco, un conjunto de estados finitos y un conjunto de transiciones entre dichos estados. Su funcionamiento se basa en una función de transición, que recibe un estado inicial y una cadena de caracteres (la cinta, la cual puede ser infinita) pertenecientes al alfabeto de entrada. La máquina va leyendo una celda de la cinta en cada paso, borrando el símbolo donde está posicionado su cabezal y escribiendo un nuevo símbolo del alfabeto de salida, para luego desplazar el cabezal a la izquierda o a la derecha (solo una celda a la vez). Esto se repite según se indique en la función de transición, para finalmente detenerse en un estado final o de aceptación, representando así la salida.

Formalmente, una máquina de Turing simple es una séptupla: $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, donde:

Q : Es un conjunto finito de estados.

Σ : Es un conjunto finito de símbolos de entrada.

Γ : Es el conjunto de símbolos de la cinta. Σ es siempre un subconjunto de Γ .

δ : La función de transición $\delta(q, X) = (p, Y, D)$, donde p es el siguiente estado en Q , Y es el símbolo en Γ que se escribe en la celda que está viendo la cabeza de la cinta y es la dirección (izq. o der.).

q_0 : Es el estado inicial.

B : Es el símbolo de vacío, que está en Γ pero no en Σ .

F : Es el conjunto de estados finales o de aceptación (Hopcroft, Motwani, & U. Ullman, 2007).

Algoritmo de una Máquina de Turing Estándar

Una vez se hayan ingresado los estados y transiciones se puede usar el método ejecutar(String cadena) para simular la Máquina de Turing, el algoritmo es el siguiente:

- Se carga la cadena en la cinta.
- Al inicio del algoritmo el estado inicial será el estado actual y el primer símbolo de la cinta será el símbolo actual.
- Si el estado actual pertenece al conjunto de los estados de aceptación entonces detener el algoritmo ya que la cadena es aceptada, sino ir al paso 4.
- Verificar si hay una transición con el estado y el símbolo actuales, en caso de no haberla detener el algoritmo ya que la cadena no es aceptada, de otro modo ir al paso 5.
- Escribir en la cinta el símbolo obtenido en la transición y mover el cabezal según la dirección

obtenida en la transición.

- El estado obtenido por la transición se vuelve el estado actual y el símbolo al que apunta el cabezal de la cinta se convierte en el símbolo actual.
- Ir al paso 3.

Creación de la Máquina de Turing Estándar

Para la creación de una Máquina de Turing Estándar se creo una instancia de la clase MT; ejemplo. Después se agregar los estados, el estado inicial, los estados finales y transiciones, en ese orden y con los métodos que se describen a continuación:

- `addEstado(int estado)` permite agregar un solo estado a la Máquina de Turing, de repetirse dicho estado no se agregará.
- `addEstados(int... estados)` permite agregar múltiples estados a la Máquina de Turing, solo se agregarán los estados que no estén en el conjunto de estados.
- `setEstadoInicial(int estado)` asigna el estado inicial de la Máquina de Turing, el estado proporcionado por el parámetro del método solo se asignará si se encuentra en el conjunto de estados.
- `addEstadoAceptacion(int estado)` agrega un estado de aceptación a la Máquina de Turing, el estado proporcionado por el parámetro del método solo se asignará si se encuentra en el conjunto de estados.
- `addEstadosAceptacion(int... estados)` agrega un estado de aceptación a la Máquina de Turing, el estado proporcionado por el parámetro del método solo se asignará si se encuentra en el conjunto de estados.
- `addTransicion(int desdeEstado, char desdeSimbolo, int haciaEstado, char haciaSimbolo, MT.Direccion direccion)` agrega una transición a la Máquina de Turing.

Interfaz gráfica

Interfaz principal:

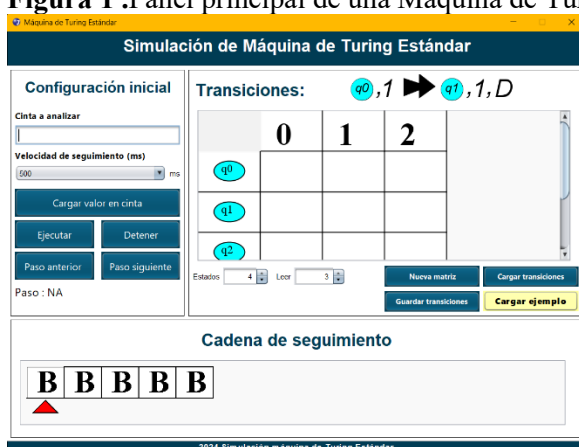
Esta figura representa una Máquina de Turing Estándar en donde se puede observar en una forma más representativa y entendible para el usuario lo siguiente; **la cinta que se ve representada en el apartado**



de cadena de seguimiento, el cabezal que podemos ver en el apartado de transiciones la cual está representada en la parte superior de la tabla con (0,1 y 2.....), el registro de estado el cual se encuentra en el mismo apartado de transiciones y está representado con (q_0 , q_1 y q_2) y la tabla.

En la información de conceptos fundamentales que se mostró anteriormente, se habla sobre que es una Máquina de Turing en un término tanto formal como informal para su mayor comprensión. Ahora se verá de manera representada en una interfaz para mayor comprensión en la figura 1.

Figura 1 .Panel principal de una Máquina de Turing Estándar.



Fuente: (elaboración propia, 2025)

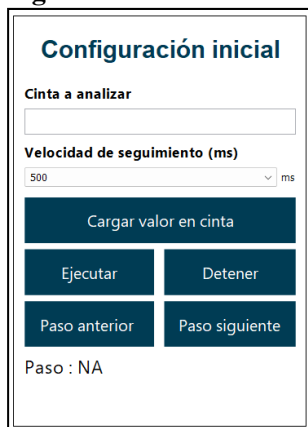
Para una mayor comprensión, vamos a dividir la interfaz principal en 3 partes. Ver figura 2.

- **Configuración inicial.**
- **Transiciones.**
- **Cadena de seguimiento.**

Donde se explicará detalladamente cada componente de cada una de ellas.

1. Interfaz: Configuración inicial.

Figura 2. Interfaz de configuración inicial.



The screenshot shows a window titled "Configuración inicial". It contains a text input field labeled "Cinta a analizar", a dropdown menu labeled "Velocidad de seguimiento (ms)" with "500" selected, and a "ms" unit label. Below these are five buttons: "Cargar valor en cinta", "Ejecutar", "Detener", "Paso anterior", and "Paso siguiente". At the bottom, it says "Paso : NA".

Fuente: (elaboración propia, 2025)

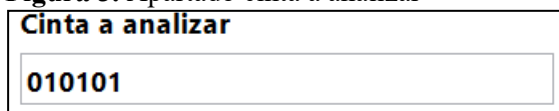
Como se observa, la configuración inicial se cuenta con 2 apartados y 5 botones, los cuales hacen una función específica, la cual se explicará a continuación.

Apartados de la configuración inicial

Cinta a analizar:

En este apartado pondrás ingresar la cadena o cinta a evaluar, ver figura 3, al darle cargar valor en la cita, en automático se reflejará en la cinta como se mostrará más adelante.

Figura 3. Apartado cinta a analizar

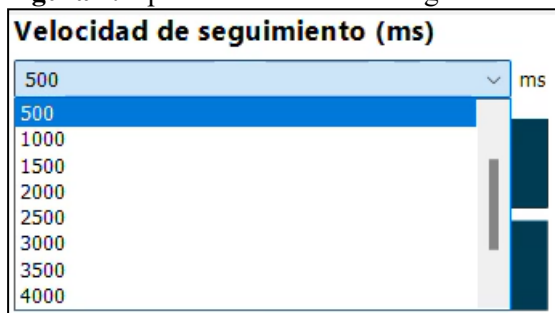


The screenshot shows a text input field with the label "Cinta a analizar" and the value "010101" entered.

Velocidad de seguimiento:

En esta parte podrás decidir qué tan lento o rápido deseas que se haga la simulación en la cinta. En donde 500 ms es el más rápido y el 4000 ms el más lento. Ver figura 4.

Figura 4. Apartado velocidad de seguimiento



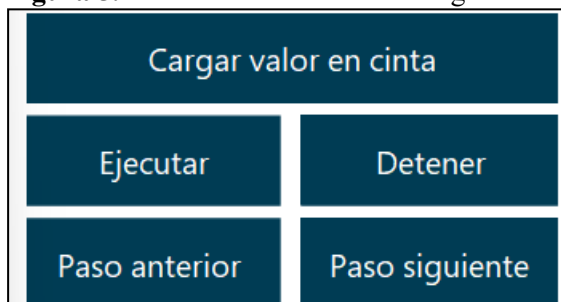
The screenshot shows a dropdown menu titled "Velocidad de seguimiento (ms)". The selected value is "500". The menu is open, showing a list of options: 500, 1000, 1500, 2000, 2500, 3000, 3500, and 4000. A vertical scrollbar is visible on the right side of the list.

Fuente: (elaboración propia, 2025)

Botones

En esta primera parte de la interfaz contamos con 5 botones, según la figura 5, los cuales su función principal es:

Figura 5. Botones de la interfaz configuración inicial.



Fuente: (elaboración propia, 2025)

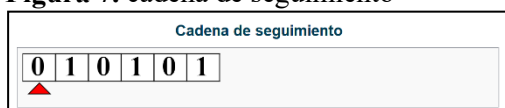
Cargar valor en la cinta: Este botón va de la mano con la figura 3, donde le ponemos la cinta o cadena que se quiera analizar. Al darle clic en automático, la cadena o cinta que ingresaste se pasará a la cinta en la parte de cadena de seguimiento, ver figura 6.

Figura 6. Cinta a analizar



Fuente: (elaboración propia, 2025)

Figura 7. cadena de seguimiento



Fuente: (elaboración propia, 2025)

Como se muestra en la figura 6 y 7, el mismo valor que se quiere analizar es el que se representa en la cinta para hacer la simulación para saber si esa cadena o cinta es aceptada o rechazada.

- **Ejecutar:** Ejecuta todo proceso tanto en la cinta como en las transiciones.
- **Detener:** Detiene todo el proceso tanto en la cinta como en las transiciones.
- **Paso anterior:** pasa al proceso anterior tanto en la cinta como en las transiciones.
- **Paso siguiente:** pasa al proceso siguiente tanto en la cinta como en las transiciones.

Interfaz: Transiciones

En esta parte es donde más interactúas con el ejecutable, ya que aquí es donde comienzas a crear la

Máquina de Turing; es aquí donde vas a poder darle los valores tanto en el cabezal como en el registro de estados, ver figura 8.

Figura 8. Interfaz: transiciones

Fuente: (elaboración propia, 2025)

Para un mayor entendimiento, vamos a presentar cómo trabajar esta parte paso a paso y qué se debe hacer en cada uno de ellos.

Botón de estados y lectura de valores

Como primer paso, lo que se hará es decirle a la Máquina cuantos estados vas a necesitar y cuantos valores se van a leer, observe figura 9.

Figura 9. Agregar estados y leer valores

Fuente: (elaboración propia, 2025)

Como se observa en la figura 9, en este caso se está solicitando 3 estados y se van a leer 3 valores. Lo cual se ve reflejado en la tabla. Tenemos 3 estados los cuales son q0, q1 y q2 y vamos a leer 3 valores que son 0,1 y ese guion significa que tienes que agregarle un dato, puede ser 0,1 o B que esa letra significa blanco.

Figura 10. Ejemplo de agregación de estados y leer

	0	1	-
q0			
q1			
q2			

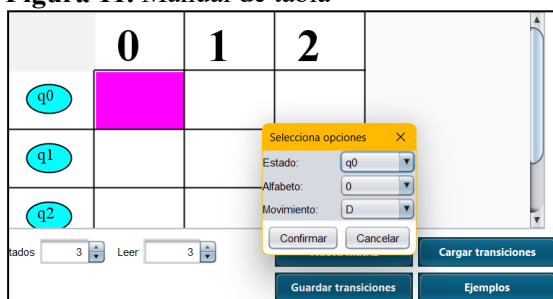
Fuente: (elaboración propia, 2025)

Tabla de transiciones

Esta parte es muy importante y clave para poder usarla. A continuación se explicará qué se tiene que hacer en la tabla.

Al darle clic en cualquier cuadrado en blanco, vamos a ingresar el valor que le vamos a dar a la Máquina para su simulación.

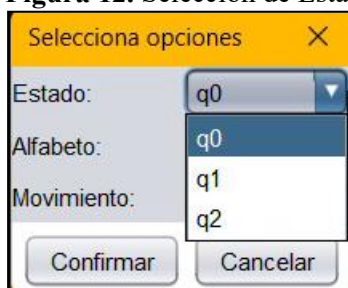
Figura 11. Manual de tabla



Fuente: (elaboración propia, 2025)

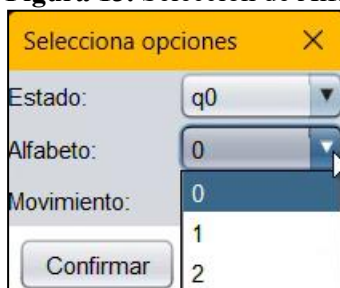
Como pueden observar en la figura 11, se le dio clic en el estado q_0 y en el valor 0, nos abrió una venta en la cual cuenta con tres opciones: la cual es estado, alfabeto y movimiento. Estovaría según los valores que se agreguen en los estados y en el cabezal.

Figura 12. Selección de Estado



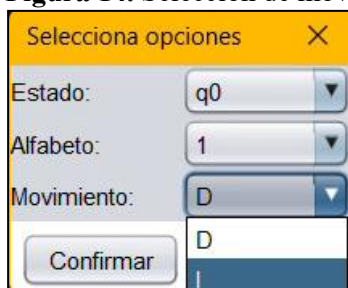
Fuente: (elaboración propia, 2025)

Figura 13. Selección de Alfabeto



Fuente: (elaboración propia, 2025)

Figura 14. Selección de movimientos



Fuente: (elaboración propia, 2025)

Este paso se repite las veces que sea necesario según los valores que se hayan ingresado, posteriormente a esto. Una vez seleccionados esos valores, se cargarán en el aparato donde se le dio clic, como se muestra en la figura 15.

Figura 15. Representación de cuando se cargan los valores en la tabla

	0	1	2
q0	(q0,1,D)		
q1			
q2			

Fuente: (elaboración propia, 2025)

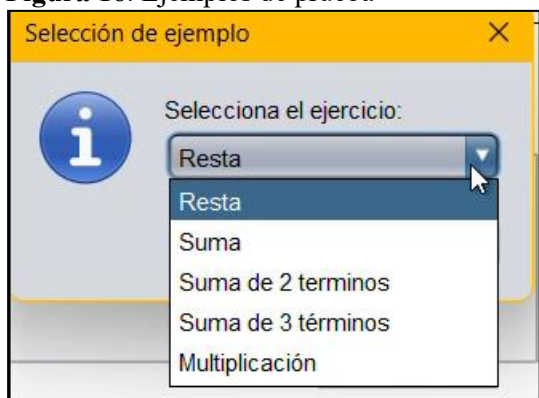
Repetir el procedimiento anterior las veces que sean necesarias o que el problema a resolver lo solicite.

Botones

Finalmente, es este apartado igual, contamos con botones, los cuales su función es:

- **Nueva matriz:** Genera una nueva matriz.
- **Guardar transiciones:** Guarda las transiciones para tenerlas como evidencia.
- **Cargar transiciones:** Carga de tu dispositivo alguna transición guardada.
- **Cargar ejemplos:** Se cuenta con 5 ejemplos para que pueda interactuar con ellos para pruebas.

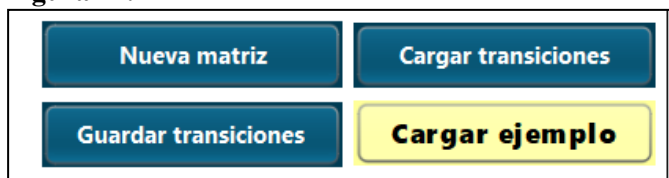
Figura 16. Ejemplos de prueba



Fuente: (elaboración propia, 2025)

Después de seleccionar alguno, le damos en el botón ok y en automático se cargarán los valores en la tabla.

Figura 17. Botones de la interfaz de transiciones



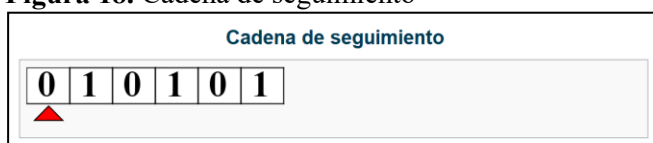
Fuente: (elaboración propia, 2025)

Interfaz cadena de seguimiento

Cinta

Finalmente, en el apartado 3, como ya se mencionó anteriormente en la figura 3, aquí se carga y se ve la simulación del recorrido de la cinta.

Figura 18. Cadena de seguimiento



Fuente: (elaboración propia, 2025)

METODOLOGÍA

Tabulación y análisis de datos en investigación

La tabulación y análisis de datos en investigación son elementos fundamentales para obtener resultados precisos y confiables. Estas técnicas permiten organizar la información recolectada de manera eficiente, identificar patrones, establecer conclusiones y realizar inferencias.

Importancia de la tabulación de datos en la investigación

- **Organizar la información:** La tabulación permite organizar grandes cantidades de datos en un formato estructurado, lo que facilita su análisis y comprensión.
- **Detectar errores y outliers:** La tabulación de datos también puede ayudar a identificar datos incorrectos o anomalías, conocidas como outliers. Esto es esencial para garantizar la calidad y confiabilidad de los resultados de la investigación.
- **Identificar patrones y tendencias:** Al tabular los datos, es posible identificar patrones y tendencias que pueden no ser evidentes a simple vista. Estos patrones pueden proporcionar información valiosa y ayudar en la toma de decisiones.

Métodos y herramientas para tabulación y análisis de datos

Existen varios métodos y herramientas disponibles para llevar a cabo la tabulación y análisis de datos en investigación. A continuación, se presentan algunos de los más utilizados:

Tabulación manual: Consiste en ingresar los datos en tablas y matrices de manera manual, ya sea en papel o en una hoja de cálculo. Esta opción puede ser adecuada para proyectos pequeños con pocos datos.

Tabla de conteo: Una tabla de conteo es una tabla con marcas de conteo para mostrar un conjunto de datos valiosos. Una tabla de conteo es un método para recopilar datos con marcas de conteo. Las marcas de conteo son frecuencias, ocurrencias o números totales que se miden para una categoría específica en un conjunto de datos. Los gráficos de conteo se utilizan en todo el mundo y son excelentes representaciones visuales de observaciones grupales.

Este trabajo de investigación se encuentra enfocado en desarrollo de simulación de MT, esto para su implementación y el desarrollo de un programa de software que facilite al estudiante que presente la necesidad de trabajar con esta materia. Para lograr la creación del software se dividirá en las siguientes tres fases: análisis teórico, desarrollo y pruebas.

Análisis teórico

En esta fase se hizo una revisión de los elementos teóricos necesarios para completar la investigación y desarrollar el programa de software que se busca como producto. Esta revisión de elementos teóricos corresponde a los diferentes algoritmos o métodos de simulación que se encuentran desarrollados, una



comparación entre ellos y la decisión de cual se usará, pudiéndose optar por el diseño de uno nuevo si el comparativo arroja inconsistencias.

Desarrollo

En esta fase se convirtió el algoritmo que se seleccionó en la etapa anterior, al lenguaje de programación que se usó, así como el diseño de la aplicación para finalmente, la implementación del algoritmo dentro de un programa usable.

Pruebas

Esta fase corresponde al testeo de la aplicación sobre el desempeño de esta en múltiples áreas, teniendo en cuenta dentro de las pruebas al usuario final que puso en uso el software para la facilitación de su aprendizaje.

RESULTADOS Y DISCUSIÓN

Objetivos logrados

Prueba piloto

Una vez obtenido un programa estable se realizó una prueba piloto ante un grupo de estudiantes cursando la asignatura de Lenguajes y Autómatas I. Para la prueba pilotos se les mostró el software a los estudiantes y se les explico lo que es una Máquina de Turing Bidireccional y sus conceptos fundamentales , aplicando ésta a 18 estudiantes de las materias de Lenguajes y Autómatas , de la carrera de Ingeniería en Sistemas Computacionales Al finalizar la prueba piloto se aplicó una encuesta para recibir retroalimentación y críticas constructivas para la mejora de los algoritmos y el entorno gráfico las cuales consto de las siguientes preguntas:

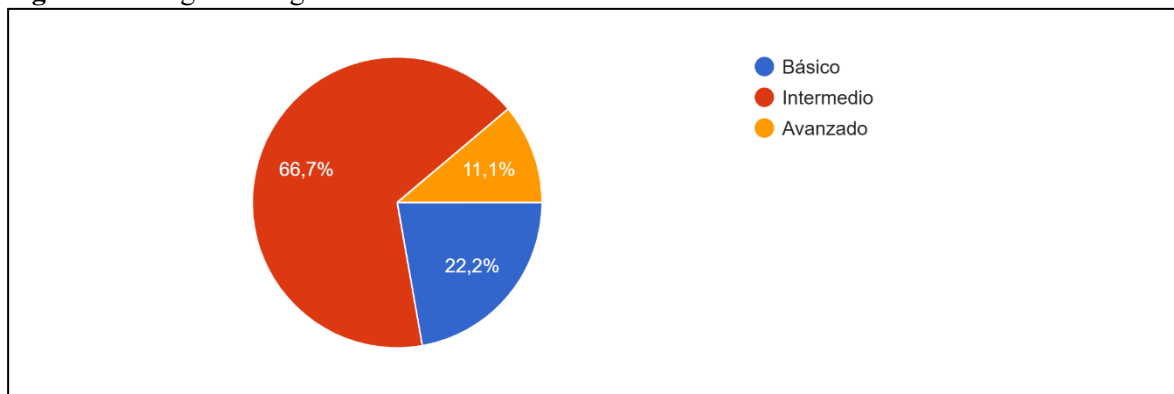
¿Cuál considera que es su nivel de comprensión sobre el concepto de Máquina de Turing Estándar?

Tabla 1 Pregunta #1

Respuesta	F	%
Básico	4	22%
Intermedio	12	67%
Avanzado	2	11%
Total	18	100%

Fuente: directa.

Figura 19. Pregunta #1 gráfica



Fuente: (elaboración propia, 2025)

Interpretación: El nivel de comprensión de los temas de Máquinas de Turing Estándar, se consideran en un nivel intermedio, con un 66.7%

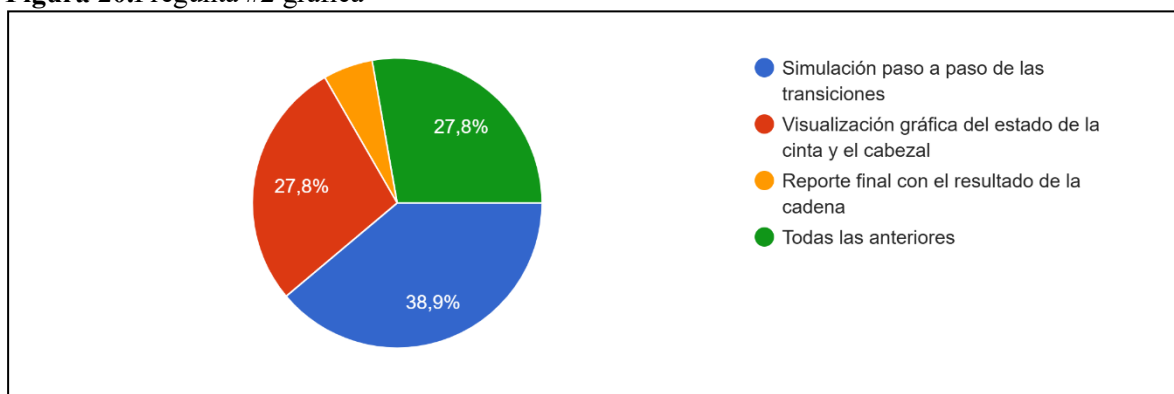
¿Qué funcionalidad le es más útil en la simulación de una Máquina de Turing Estándar?

Tabla 2. Pregunta #2

Respuesta	F	%
Simulación paso a paso de las transiciones	7	39%
Visualización gráfica del estado de la cinta y el cabezal	5	28%
Reporte final con el resultado de la cadena	1	5%
Todas las anteriores	5	28%
Total	18	100%

Fuente: directa.

Figura 20. Pregunta #2 gráfica



Fuente: (elaboración propia, 2025)

Interpretación: Un significativo 38.9% consideran la simulación paso a paso como la de mayor utilidad.

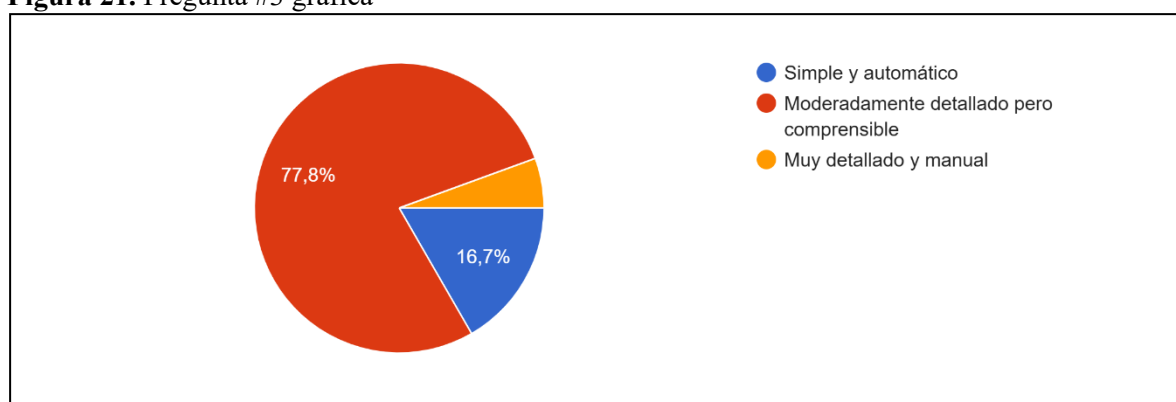
¿Cómo considera que debería ser el proceso para definir las reglas de transición en el algoritmo?

Tabla 3. Pregunta #3

Respuesta	F	%
Simple y automático	3	17%
Moderadamente detallado pero comprensible	14	78%
Muy detallado y manual	1	5%
Total	18	100%

Fuente: directa.

Figura 21. Pregunta #3 gráfica



Fuente: (elaboración propia, 2025)

Interpretación: El 77.8% considera que el proceso de definición de las reglas es moderadamente detallado, pero comprensible.

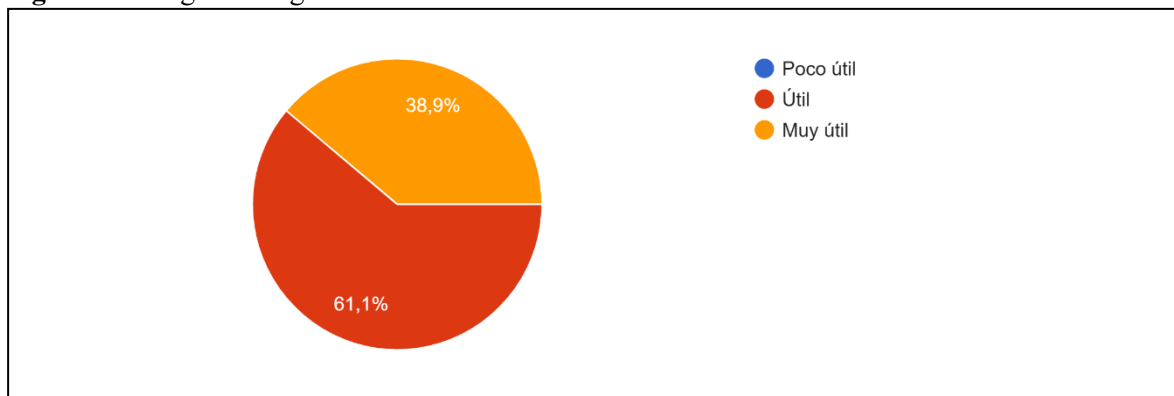
¿Qué tan útil considera la opción de pausar la simulación para observar cada paso del procesamiento?

Tabla 4. Pregunta #4

Respuesta	F	%
Poco útil	0	0%
Útil	11	61%
Muy útil	7	39%
Total	18	100%

Fuente: directa.

Figura 22. Pregunta #4 gráfica



Fuente: (elaboración propia, 2025)

Interpretación: Una mayoría significativa la considera muy útil, en tanto la gran mayoría está de acuerdo en que es útil.

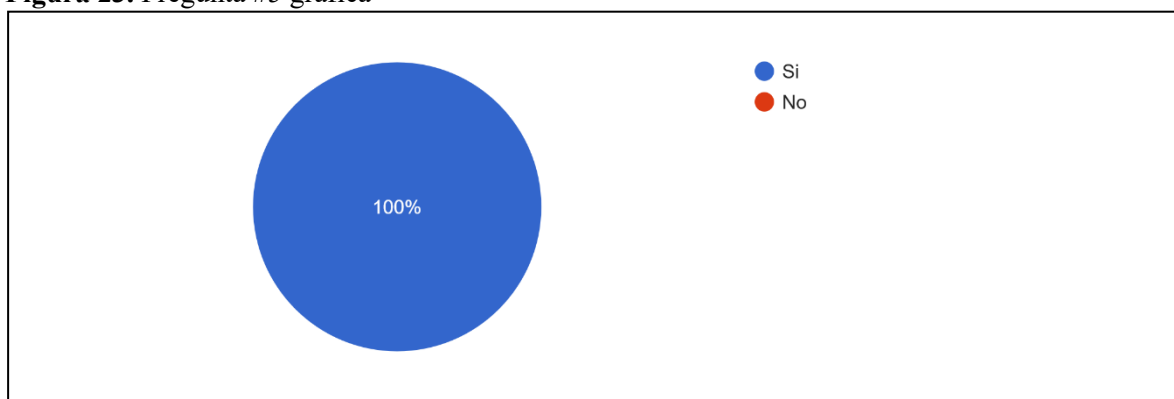
¿Le gustaría que el algoritmo permita configurar los estados iniciales, finales y reglas específicas?

Tabla 5. Pregunta #5

Respuesta	F	%
Si	18	100%
No	0	0%
Total	18	100%

Fuente: directa.

Figura 23. Pregunta #5 gráfica



Fuente: (elaboración propia, 2025)

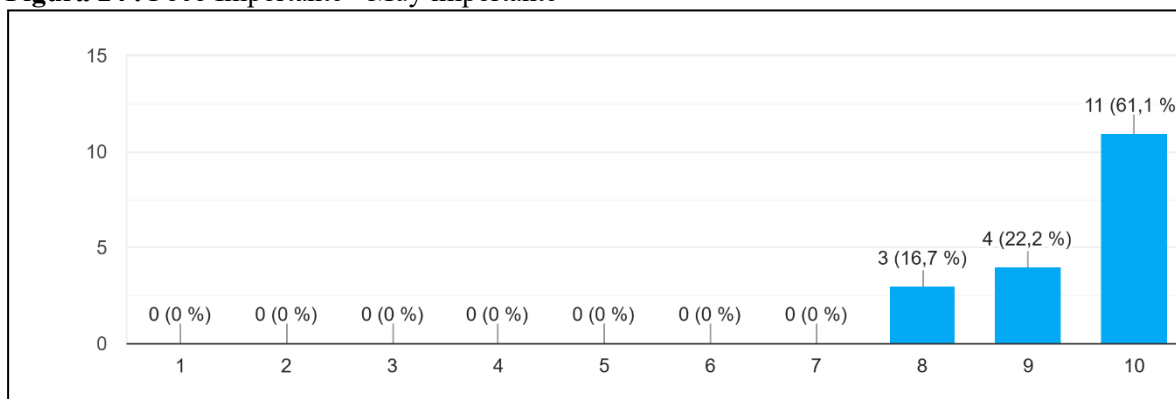
Interpretación: Todos los encuestados están de acuerdo en que se debe permitir la configuración de estados iniciales, finales y reglas específicas.

¿Cómo considera la precisión en las transiciones de una Máquina de Turing? Califica del 1 al 10, donde 1 es calificación baja y 10 calificación alta.

Tabla 6. Pregunta #6 califica del 1 al 10

	1	2	3	4	5	6	7	8	9	10	
Poco importante								3	4	11	Muy importante
Ineficiente							3	4	2	9	Eficiente
Lento							3	4	2	9	Rápido

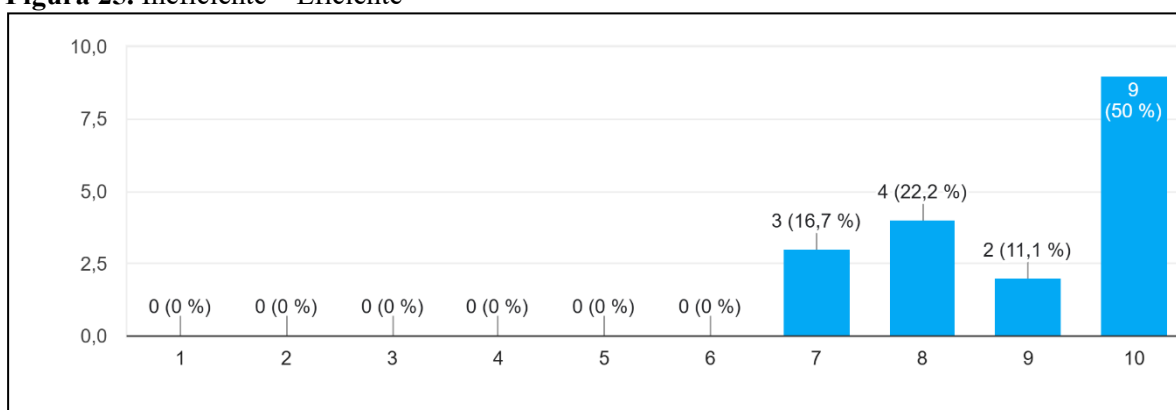
Figura 24 . Poco Importante - Muy importante



Fuente: (elaboración propia, 2025)

Interpretación: La precisión en las transiciones de una MTS se considera muy importante entre los puntos 8, 9 y 10, con la totalidad de los encuestados.

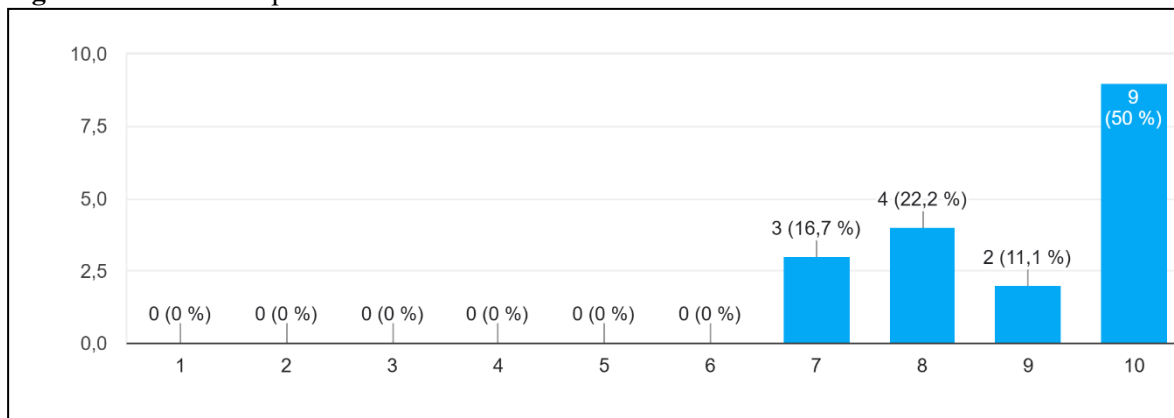
Figura 25. Ineficiente – Eficiente



Fuente: (elaboración propia, 2025)

Interpretación: En cuanto a la eficiencia, los encuestados se manifiestan entre los puntos 7, 8, 9 y 10, cargado hacia el adjetivo eficiente.

Figura 26. Lento – Rápido



Fuente: (elaboración propia, 2025)

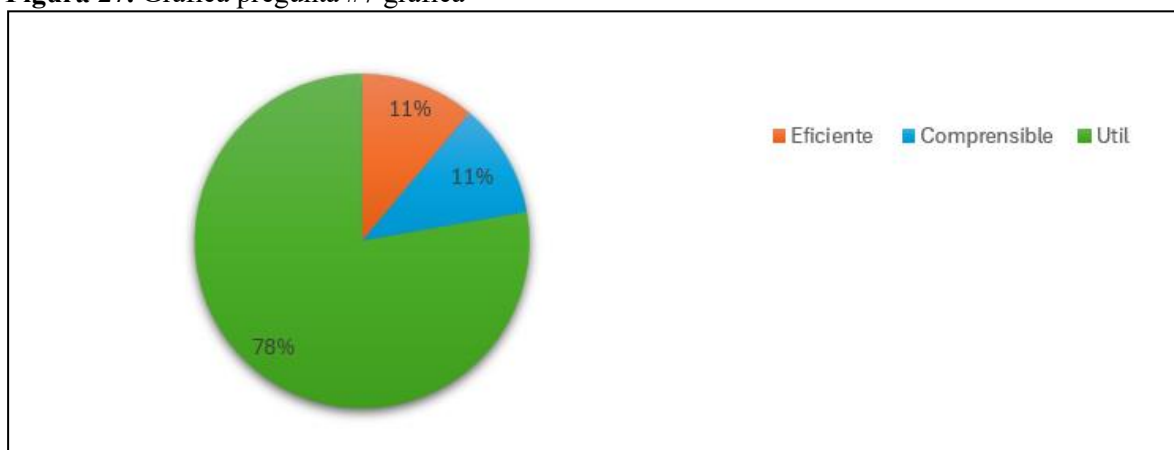
¿Cómo consideras el apoyo de un algoritmo que solucione problemas computables a través de una Máquina de Turing gráfica?

Tabla 7. Pregunta #7

Respuesta	F	%
Eficiente	2	11%
Comprensible	2	11%
Útil	14	78%
Total	18	100%

Fuente: directa.

Figura 27. Gráfica pregunta #7 gráfica



Fuente: (elaboración propia, 2025)

Interpretación: Se destaca que 78% menciona que contar con la herramienta da una comprensión. Al tener representaciones gráficas, los estudiantes pueden asimilar mejor el funcionamiento de la Máquina de Turing.

CONCLUSIÓN

La investigación presentada culmina con el desarrollo exitoso de un software que simula una Máquina de Turing Estándar, diseñado específicamente para estudiantes de Ingeniería en Sistemas Computacionales. Este programa no solo facilita la comprensión de conceptos fundamentales relacionados con lenguajes formales y modelos matemáticos, sino que también refuerza el razonamiento lógico-matemático requerido para desarrollar compiladores, objetivo principal de la materia "Lenguajes y Autómatas I".

La metodología utilizada, combinando investigación teórica, diseño e implementación de algoritmos y pruebas piloto con usuarios finales, permitió garantizar que la herramienta sea funcional, intuitiva y adecuada para el aprendizaje. Los resultados de la encuesta aplicada a estudiantes confirman la utilidad del software, destacando su capacidad para visualizar de manera gráfica y comprensible el funcionamiento de las Máquinas de Turing, lo que facilita su asimilación.

En conclusión, este proyecto demuestra cómo una herramienta tecnológica puede integrarse de manera efectiva en la educación, contribuyendo significativamente al aprendizaje de temas complejos en ciencias computacionales y potenciando la capacidad de los estudiantes para enfrentar problemas computacionales.

REFERENCIAS BIBLIOGRÁFICAS

- A.V.Aho. (2008). *compiladores: principios, técnicas y herramientas*.
- Alcance, I. I. (2023). *Metodología de Prototipo para el Desarrollo del Software*. TecnoDigital.
- Díaz Leal Guzmán, H. (2020). *Lenguajes y autómatas. Teoría de grupos. Teoría de anillos. Bases de Grobner*. Universidad Autónoma Metropolitana, Unidad Azcapotzalco.
- Giro, J., Vázquez, J., & Meloni, B. (2015). *Lenguajes formales y teorías de autómatas*. Marcombo.
- Giró, J., Vázquez, J., Meloni, B., & Constable, L. (2015). *Lenguajes formales y teoría de autómatas*. Alfaomega Grupo Editor.
- Hopcroft, J. E., Motwani, R., & Ullman, J. (2007). *Introducción a la teoría de Autómatas, Lenguajes y Computación* (Vol. 2a Edición). Madrid, España: PEARSON Addison Wesley.
- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). *Teoría de autómatas, lenguajes y computación* (2.ª ed., traducción al español). Madrid, España: Pearson Educación.



Kelly, D. (2002). *Teoría de autómatas y lenguajes formales* (Vol. primera edición). España: PRENTICE HALL.

Lenguajes Formales y Autómatas. (2017). *Máquinas de Turing*. Recuperado de

https://lenguajesformalesyautomatas.files.wordpress.com/2017/12/maq_turing.pdf

Miranda Perea, F. E., Reyes Cabello, A. L., Reyes Sánchez, R., & González Huesca, L. C. (2013). La máquina de Turing en el ámbito de los lenguajes de programación. *Miscelánea Matemática*, 56, 145–178. Recuperado de

https://miscelaneamatematica.org/download/tbl_articulos.pdf2.9514bf757b3035ac.353630372e706466.pdf

Pérez Jiménez, M. de J. (2020). *Tema 2: Modelos de computación: Máquinas de Turing* [PDF]. Universidad

de Sevilla. Recuperado de <https://www.cs.us.es/~marper/docencia/TCC-2019-2020/temas/tema-2-trans.pdf>

TALFI. (2009). *Una herramienta de aprendizaje para Teoría de Autómatas y Lenguajes Formales*.

Turing, A. (1948). *Intelligent Machinery* (manuscript).

Universidad Nacional del Centro de la Provincia de Buenos Aires. (2009). *Máquinas de Turing*. Facultad de Ciencias Exactas. Recuperado de

<https://users.exa.unicen.edu.ar/catedras/ccomp1/ApunteMT.pdf>

Viso Gurovich, E. (2008). *Introducción a la Teoría de la Computación*. UNAM.

Vladimir, I. (2021). *La máquina de Turing*. En *Lenguajes formales y autómatas*. Recuperado de

https://ivanvladimir.gitlab.io/lfya_book/docs/08lam%C3%A1quinaconcinta/01m%C3%A1quinadeturing/

