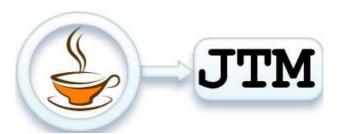


### Integrantes

Morones Flores Saúl Isaac Sandoval Delgado Juan Antonio Sena Domínguez Marcela Guadalupe



Materia: Graficación

Ing. Juan Manuel Moran Amador

Octavo Semestre

16/06/2009



### Introducción

Dentro de las ramas de la computación existe una ciencia que ha sido muchas veces explorada y no por muchos entendida, la Teoría de la Computación. Esta ciencia se basa en la computación y la matemática y centra su estudio en la definición formal de los cómputos, llamando cómputo a una serie de algoritmos que al dar una entrada de datos arrojan resultados o soluciones a un problema determinado.

La Teoría de la computación abarca muchas investigaciones, teoremas, y teorías sobre estos algoritmos y soluciones, entre ellas se encuentra la Máquina de Turing (MT) introducida por Alan Turing (1936) el cual es un modelo computacional, que estudia la cuestión planteada por David Hilbert sobre si las matemáticas son decidibles, es decir, si hay un método definido que pueda aplicarse a cualquier sentencia matemática y que nos diga si esa sentencia es cierta o no; por lo que ésta máquina es un modelo matemático abstracto que formaliza el concepto de algoritmo.

#### **JTM**

La Máquina de Turing, es pues, el enfoque de este estudio, el cual nos ha llevado al desarrollo de una aplicación **JTM** (**Java Turing Machine**) que nos permite elaborar un diagrama que represente una máquina de turing. Si bien la MT tiene una representación lógica, también le corresponde una representación gráfica, la cual plasma la representación lógica y abstracta en un diagrama que tiene símbolos y figuras específicas, además de la nomenclatura exacta de la máquina. La representación en forma de diagrama permite visualizar de una forma convencional el algoritmo que engloba la máquina abstracta y reduce la dificultad de entendimiento, abriendo un panorama al estudiante sobre este proceso.

JTM es una aplicación enfocada a plasmar estas representaciones gráficas, dotando a los usuarios de las herramientas necesarias (símbolos, figuras, lienzo, entre otras.), sirviendo así, como un complemento académico, y porque no, también una aplicación para ayudar a los investigadores en esta ciencia.

El funcionamiento básico de esta aplicación es graficar en un lienzo los elementos que el usuario desee para construir la máquina, estos elementos pueden ser: estados, transiciones entre estados, y etiquetas. Estos son los elementos básicos para la representación de una máquina de turing. El usuario puede modificar estos elementos cuantas veces lo desee, eliminarlos, crearlos y moverlos, además de tener la opción de hacer zoom. Y como toda buena aplicación estos gráficos pueden guardarse para su posterior uso y portabilidad.

JTM es en sí, un graficador de Máquinas de Turing!

## Objetivos

### Objetivo general

Representar de manera gráfica una máquina de turing (MT), haciendo uso de herramientas computacionales y permitiendo a los usuarios utilizar esta aplicación como una herramienta académica y con un enfoque de estudio. Plasmar en la aplicación diagramas elaborados por los usuarios utilizando la simbología correcta, haciendo de estos diagramas archivos portables, funcionales y entendibles.

### Objetivos específicos

- Representar gráficamente una MT
- Permitir el manejo del Zoom de las maquinas dibujadas
- Generar archivos de las maquinas elaboradas
- Ser una herramienta de uso académico
- Adentrarse en el ámbito de estudio científico
- Dar facilidad a los usuarios para crear MT bien formadas
- Dotar de herramientas a los usuarios para representar gráficamente las MT
- Hacer portables y reutilizables los archivos generados por los usuarios
- Aportar a los usuarios una aplicación Multiplataforma (Java)



# ■ Diagrama de clases de JTM

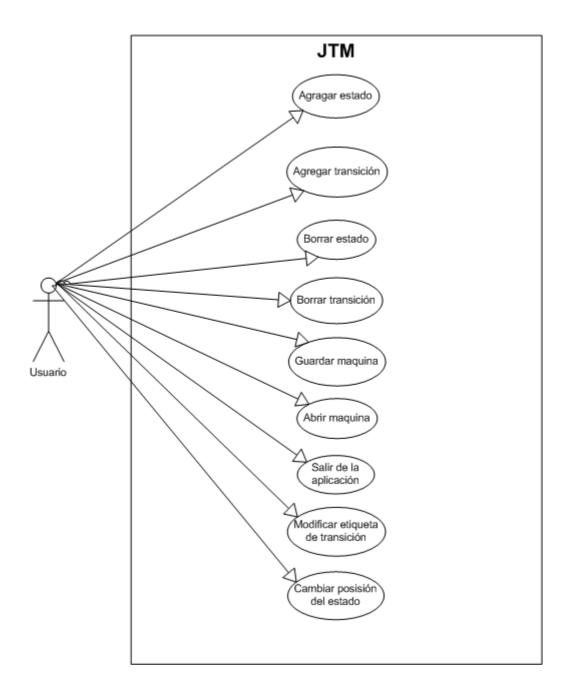




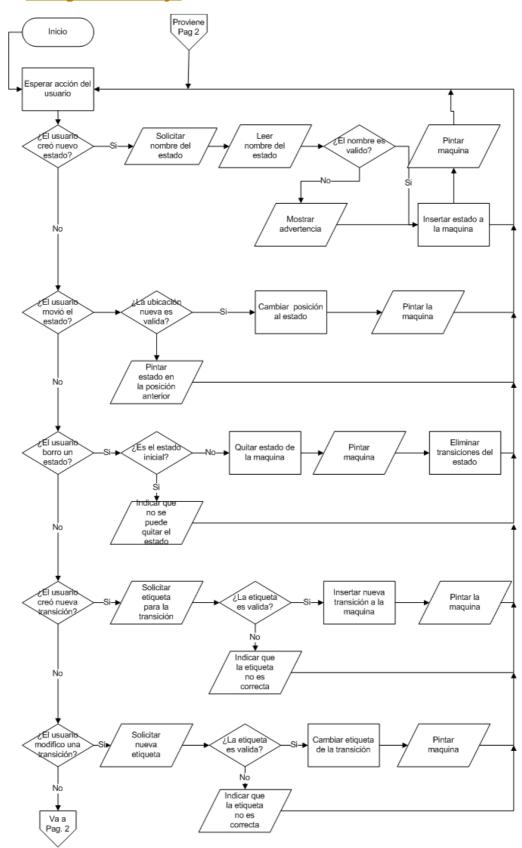




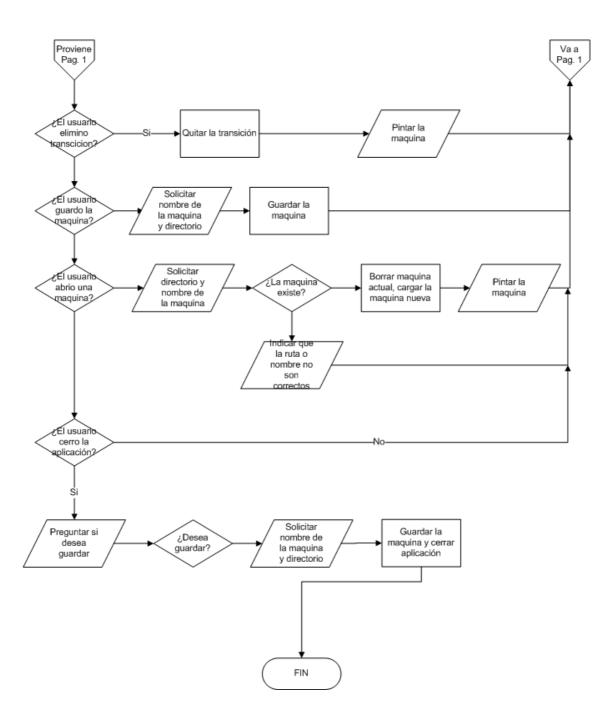
## Diagrama de casos de uso



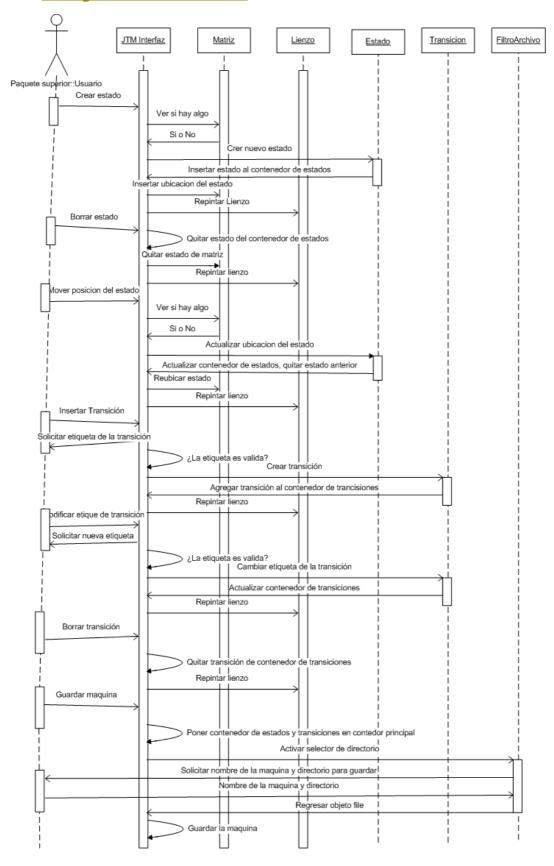
# ■ Diagrama de flujo



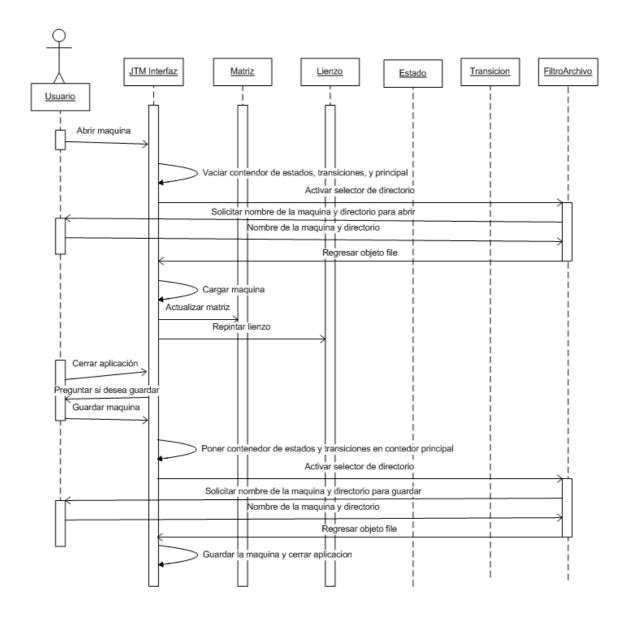
# ■ Diagrama de flujo



# Diagrama de secuencias



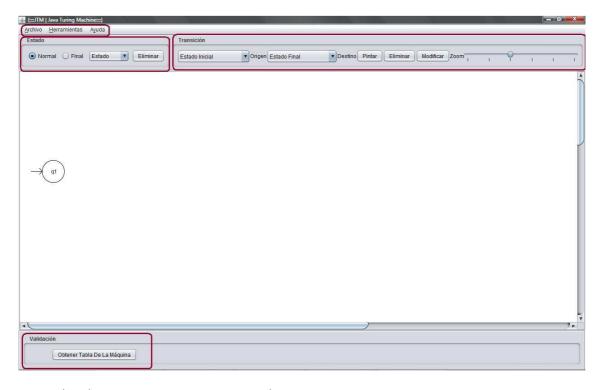
## Diagrama de secuencias





## Funcionamiento de JTM

Pantalla inicial de Java Turing Machine

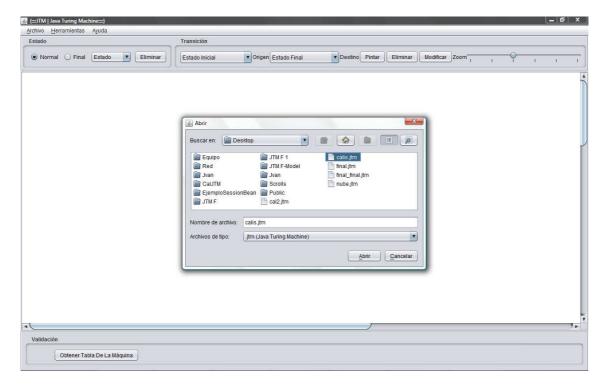


Se pueden distinguir varias secciones en el programa:

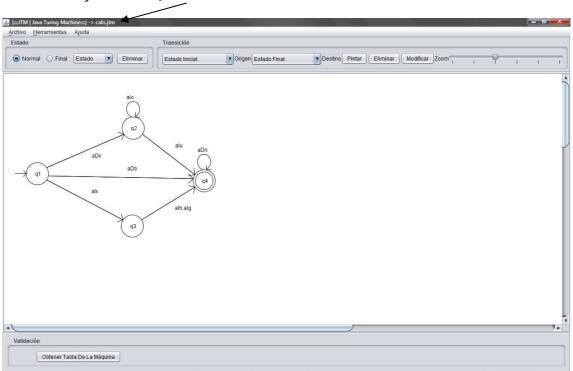
- Sección de Menú
- Sección de Estado
- Sección de Transición
- Sección de Dibujo (lienzo)
- Sección de Validación

En la sección de Menú, podemos distinguir varias opciones que son:

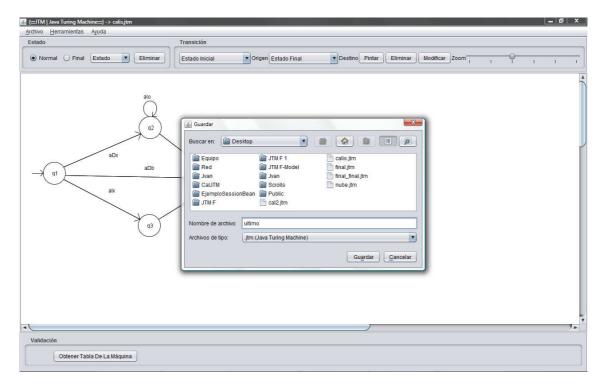
- Archivo.- Estamos utilizando la tecnología de Serialización de objetos de java para guardar y abrir archivos.
  - o Abrir .- Abre un archivo con extensión .jtm



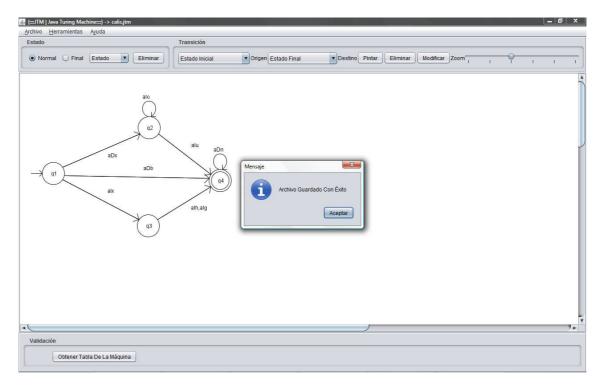
Archivo "calis.jtm" abierto, en la barra de título se muestra el nombre del archivo abierto



o Guardar.- Guarda un archivo con extensión .jtm

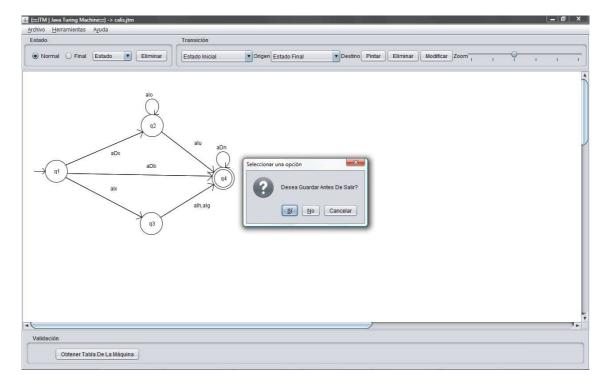


Una vez que el archivo es guardado muestra un mensaje de que el archivo ha sido guardado correctamente



o Salir.- Sale del sistema

El programa pregunta si desea guardar el archivo antes de salir, de ser sí, lo guarda, si no simplemente sale del sistema, si da clic en cancelar el sistema no realiza ninguna acción.

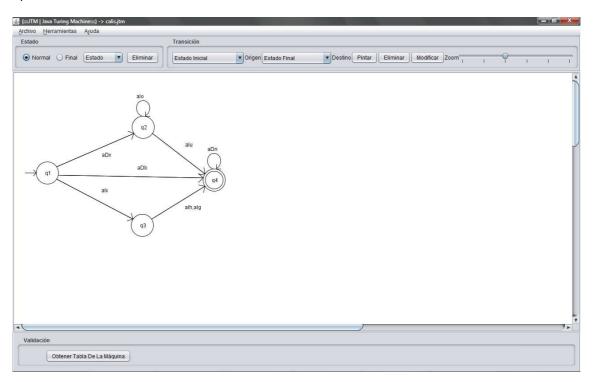


#### Herramientas

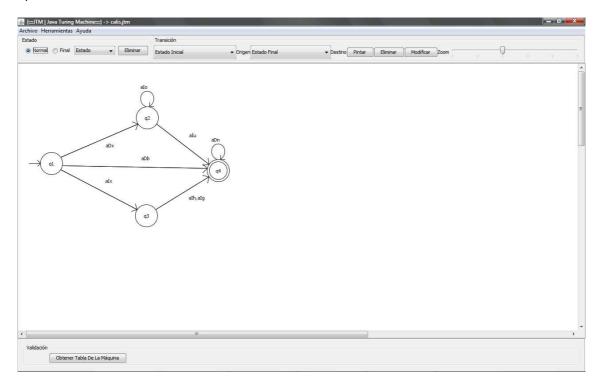
o Apariencia. - Permite elegir de entre las apariencias mostradas por el S.O.

Cuando hablamos de JTM hablamos de portabilidad, java tiene clases que permiten cambiar la apariencia de swing, de acuerdo al S.O. en el que se ejecute, JTM permite cambiar apariencias de acuerdo a los S.O. en que se ejecute.

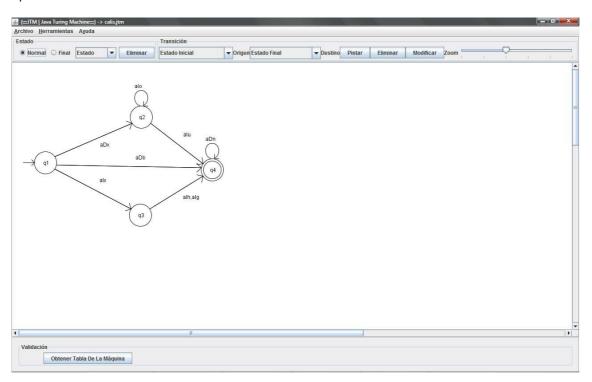
#### Apariencia Nimbus



#### Apariencia Windows

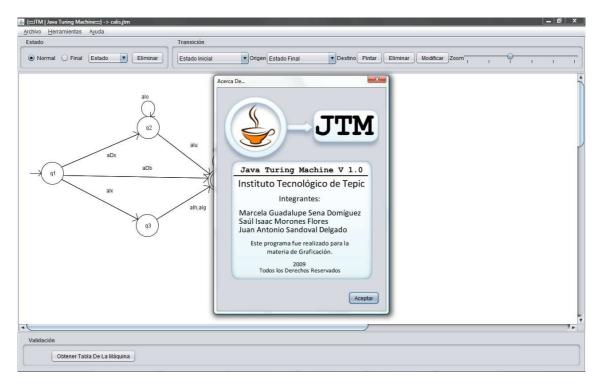


#### Apariencia Metal



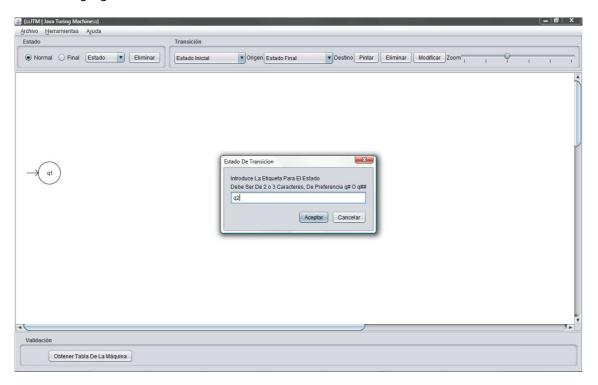
Y muchas más apariencias dependiendo del S.O.

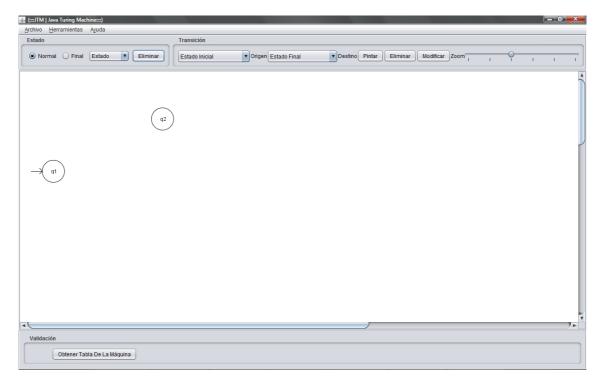
- Ayuda
  - o Acerca de.- Muestra información sobre el proyecto y sus desarrolladores.



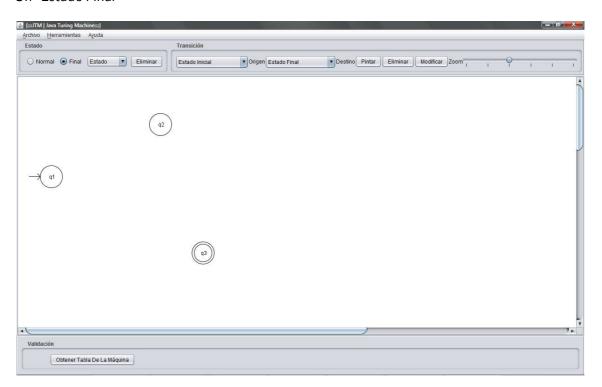
La sección de Estado, es la sección que nos permite agregar más estados a nuestra área de dibujo.

Podemos agregar un "Estado Normal"

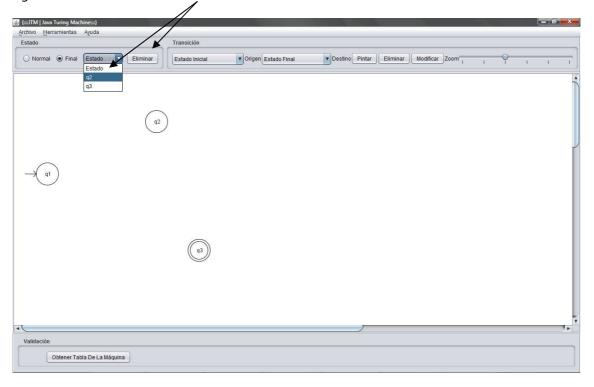


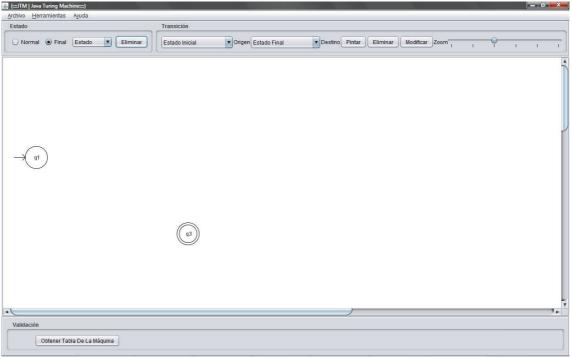


#### Un "Estado Final"



Y también podemos eliminar cualquier estado que hayamos creado, si el estado contiene alguna transición también será eliminada

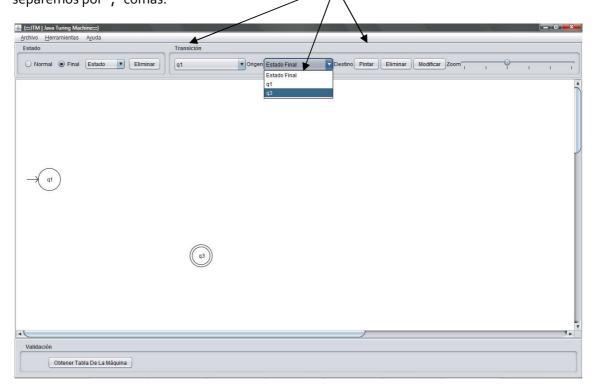


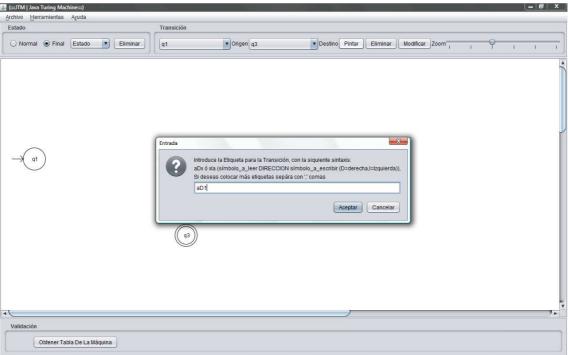


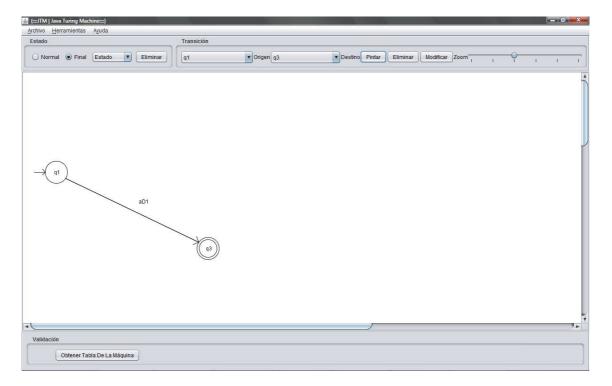
Cabe destacar que el "Estado Inicial q1" no se puede eliminar, ya que por rigor, una maquina de turing debe de contar con uno y solo un estado inicial.

En la sección de Transición podemos crear, modificar y eliminar transiciones.

Crear una transición: Debemos de elegir los Estado de origen -> Estado Destino a los que unirá nuestra transición, también tenemos que asignarle una etiqueta de tipo "Carácter Leído – Movimiento – Carácter A Escribir" por ejemplo: AD1, si encuentra una A, escribe un 1 y se mueve a la Derecha, podemos asignar tantas etiquetas como queramos, siempre y cuando las separemos por "," comas.



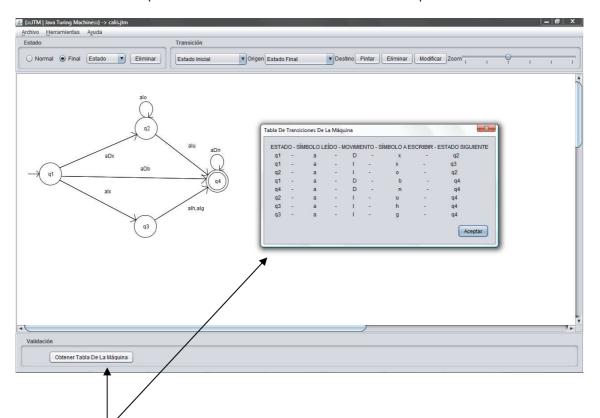




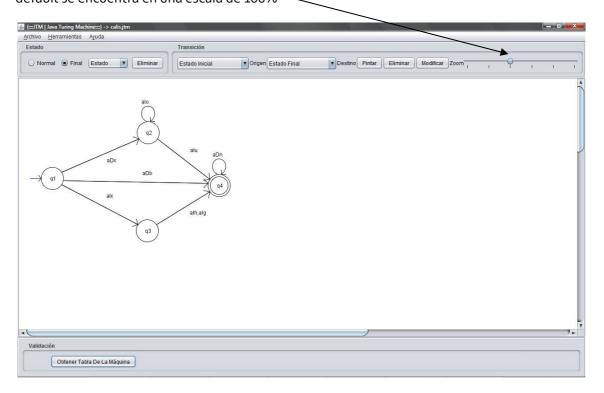
Podemos modificar cualquier etiqueta que tengamos, solo seleccionamos el estado de origen y estado destino de la etiqueta y presionamos el botón de modificar

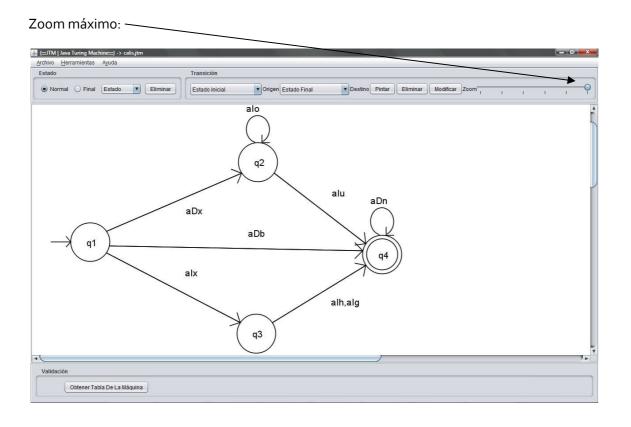
De igual manera podemos eliminar la transición que seleccionemos.

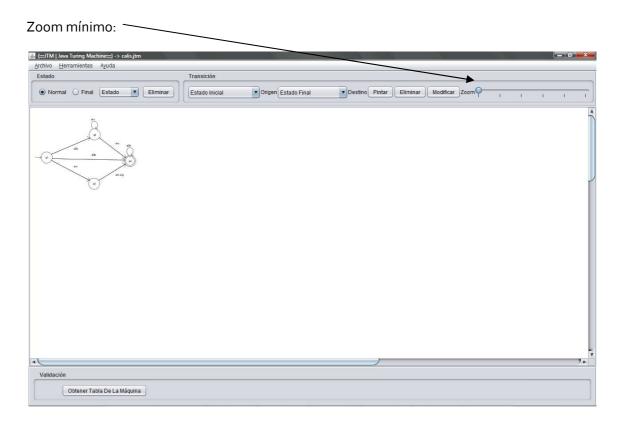
JTM permite visualizar la tabla de transiciones generada por la maquina que dibujemos, para esto solo debemos de presionar el botón "Obtener Tabla De La Máquina"



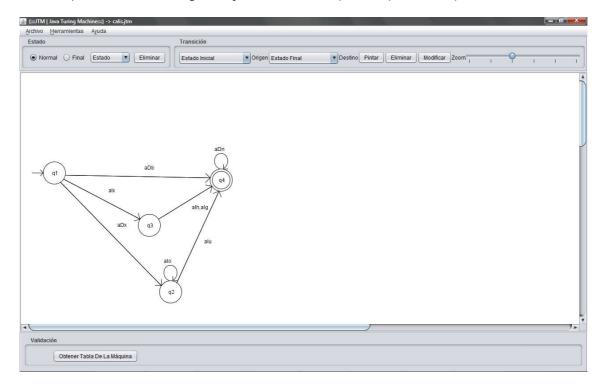
Por último, JTM permite manejar un Zoom sobre el lienzo y podemos agrandar o disminuir la maquina que dibujemos, para esto, solo es cuestión de desplazar la barra de zoom, que por default se encuentra en una escala de 100%

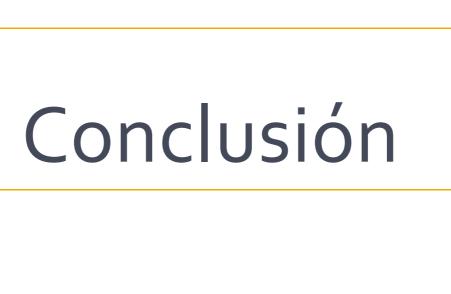






JTM también permite manipular un estado sobre el lienzo, aunque este funcionamiento es difícil de plasmar en una imagen, dejamos al usuario que lo experimente por sí mismo.





## Conclusión

Solo queda agregar que el JTM satisface las necesidades que se nos fueron requeridas en la materia de Graficación, las cuales fueron:

• Modelador gráfico de maquinas virtuales en java o flash.

Nosotros decidimos hacerlo en java por ser un lenguaje bastante potente y robusto, así como su capacidad de ser multiplataforma (JTM también funciona en Linux y Solaris, se probó y funcionó).

• La aplicación tiene que manejar un Zoom de los modelos que realice.

Efectivamente JTM permite al usuario utilizar el zoom y visualizar sus maquinas de turing en distintas escalas.

• La aplicación debe de poder guardar y abrir archivos de los modelos que haga.

La serialización de objetos en java es algo muy común, JTM hace uso de esta tecnología y así es como JTM permite al usuario guardar en archivos sus maquinas y poder abrirlas después, independientemente del S.O. en el que se esté trabajando

 La aplicación tiene que mostrar la tabla de transiciones generada por la maquina que se haya realizado.

JTM muestra la descomposición de la máquina de turing plasmada en el lienzo en la tabla de transiciones que permite validar una maquina de turing

Por último y más importante, JTM fue desarrollado única y exclusivamente por los integrantes del equipo, todas las líneas de código son de nuestra autoría.

El proyecto fue realizado en Netbeans versión 6.5 y se consultó el libro "Como Programar en Java" 5ta Edición para referencias y apoyo.