

Nombre de la práctica	AUTOMATAS FINITOS-U3			No.	1
Asignatura:	LENGUAJES Y AUTÓMATAS I	Carrera:	INGENIERÍA EN SISTEMAS COMPUTACIONALES 3501	Duración de la práctica (Hrs)	5 horas

**NOMBRE DEL ALUMNO: Raúl Ciriaco Castillo**  
**GRUPO: 3501**

## I. Competencia(s) específica(s):

Crea y reconoce autómatas finitos en un lenguaje de programación para la solución de un problema.

**Encuadre con CACEI:** Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en la materia.

No. atributo	Atributos de egreso del PE que impactan en la asignatura	No. Criterio	Criterios de desempeño	No. Indicador	Indicadores
2	El estudiante diseñará esquemas de trabajo y procesos, usando metodologías congruentes en la resolución de problemas de Ingeniería en Sistemas Computacionales	CD1	Identifica metodologías y procesos empleados en la resolución de problemas	I1	Identificación y reconocimiento de distintas metodologías para la resolución de problemas
		CD2	Diseña soluciones a problemas, empleando metodologías apropiadas al área	I1	Uso de metodologías para el modelado de la solución de sistemas y aplicaciones
				I2	Diseño algorítmico (Representación de diagramas de transiciones)
3	El estudiante plantea soluciones basadas en tecnologías empleando su juicio ingenieril para valorar necesidades, recursos y resultados esperados.	CD1	Emplea los conocimientos adquiridos para el desarrollar soluciones	I1	Elección de metodologías, técnicas y/o herramientas para el desarrollo de soluciones
				I2	Uso de metodologías adecuadas para el desarrollo de proyectos
				I3	Generación de productos y/o proyectos
		CD2	Analiza y comprueba resultados	I1	Realizar pruebas a los productos obtenidos
				I2	Documentar información de las pruebas realizadas y los resultados

## II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Laboratorio de cómputo y equipo de cómputo personal.

## III. Material empleado:

- Equipo de cómputo
- Software para desarrollo:
  - Draw Graph (diagramas de transición)
  - Excel (tablas de transición)

## IV. Desarrollo de la práctica:

### EJERCICIO 1:

1. Genera un autómata que permita conocer números enteros y decimales para reconocer operaciones aritméticas básicas (suma, resta, multiplicación, división) así también como la raíz cuadrada y potencia. El último símbolo que se genere en la cadena o cadenas deberá considerar el (=) y designación del resultado.
2. Genera la definición formal, expresión regular, lenguaje por comprensión.
3. Convierte el diagrama a un autómata finito determinístico y genera la definición formal.

#### - Definición Formal:

$$AFN < S, \Sigma, p, i, F >$$

Donde:

- **S** = Conjunto finito de estados
- **$\Sigma$**  = Alfabeto de la maquina
- **P** = Subconjunto de  $s^* \Sigma$  llamado relación de Transición
- **i** = Un elemento de S llamado estado inicial
- **F** = Subconjunto de S llamados estados de aceptación.

$$S = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_{17}, q_{18}, q_{19}, q_{20}\}$$

$$\Sigma = \{\text{operador, raiz, potencia, numero, punto, igual, negativo}\}$$

$$\text{operador} = \{+, -, *, /\}$$

$$\text{raiz} = \{\text{sqrt}\}$$

$$\text{potencia} = \{\wedge\}$$

$$\text{numero} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{punto} = \{.\}$$

$$\text{igual} = \{=\}$$

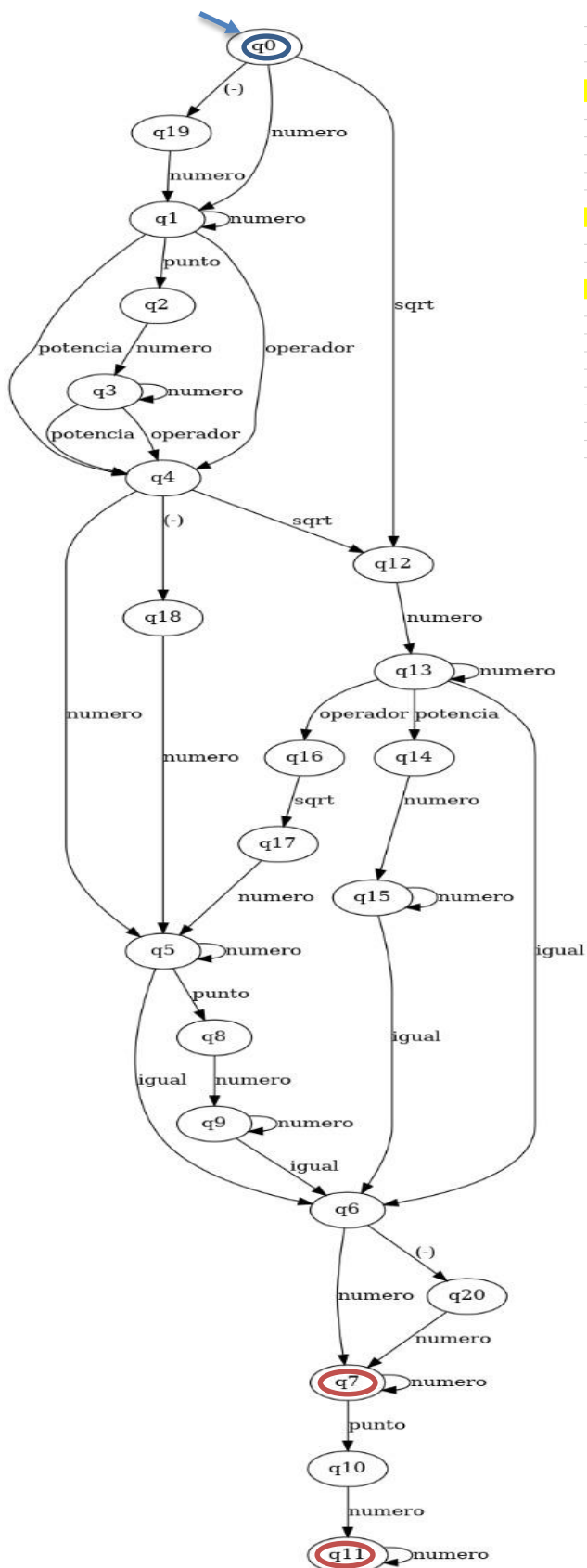
$$\text{negativo} = \{(-)\}$$

$$i = \{q_0\}$$

$$F = \{q_7, q_{11}\}$$

#### P: Puede estar representada de dos maneras:

$$P = \{(q_0, \text{numero}, q_1), (q_0, (-), q_{19}), (q_0, \text{sqrt}, q_{12}), (q_1, \text{numero}, q_1), (q_1, \text{punto}, q_2), (q_1, \text{operador}, q_4), (q_1, \text{potencia}, q_4), (q_2, \text{numero}, q_3), (q_3, \text{numero}, q_3), (q_3, \text{operador}, q_4), (q_3, \text{potencia}, q_4), (q_4, \text{numero}, q_5), (q_4, (-), q_{18}), (q_4, \text{sqrt}, q_{12}), (q_5, \text{numero}, q_5), (q_5, \text{punto}, q_8), (q_5, \text{igual}, q_6), (q_6, \text{numero}, q_7), (q_6, (-), q_{20}), (q_7, \text{numero}, q_7), (q_7, \text{punto}, q_{10}), (q_8, \text{numero}, q_9), (q_9, \text{numero}, q_9), (q_9, \text{igual}, q_6), (q_{10}, \text{numero}, q_{11}), (q_{11}, \text{numero}, q_{11}), (q_{12}, \text{numero}, q_{13}), (q_{13}, \text{numero}, q_{13}), (q_{13}, \text{operador}, q_{16}), (q_{13}, \text{potencia}, q_{14}), (q_{13}, \text{igual}, q_6), (q_{14}, \text{numero}, q_{15}), (q_{15}, \text{numero}, q_{15}), (q_{15}, \text{igual}, q_6), (q_{16}, \text{sqrt}, q_{17}), (q_{17}, \text{numero}, q_5), (q_{18}, \text{numero}, q_5), (q_{19}, \text{numero}, q_1), (q_{20}, \text{numero}, q_7)\}$$



EJERCICIO 1									
TABLA DE TRANSICIONES									
	numero	negativo	punto	operador	raiz	potencia	igual		
inicial	q0	q1	q19						3
	q1	q1		q2	q4		q4		4
	q2	q3							1
	q3	q3			q4		q4		3
	q4	q5	q18			q12			3
	q5	q5		q8				q6	3
	q6	q7	q20						2
aceptacion	q7	q7		q10					2
	q8	q9							1
	q9	q9						q5	2
	q10	q11							1
aceptacion	q11	q11							1
	q12	q13							1
	q13	q13							1
	q14	q15			q16		q14	q6	4
	q15	q15							1
	q16	q17				q17			2
	q17	q5							1
	q18	q5							1
	q19	q1							1
	q20	q7							1
Total Transiciones									39

## Lenguaje por Comprensión:

$L = \{w \in \{\text{operador, raiz, potencia, numero, punto, igual, negativo}\}^* \mid w \text{ cumple con } (((\text{numero} | (-)\text{numero}^+ (\text{operador} | ^ | (\text{punto})\text{numero}^+ (\text{operador} | ^)) (\text{sqrt}(\text{numero}^+) \text{operador} (\text{sqrt}(\text{numero}^+ | ^\text{numero}^+ \text{igual} | \text{igual}) | (-)\text{numero}^+ \text{punto}(\text{numero}^+) \text{igual} | \text{numero}^+ \text{igual})(\text{numero}^+ | (-)\text{numero}^+ | \text{numero}^+ \text{punto}(\text{numero}^+) | (-)\text{numero}^+ \text{punto}(\text{numero}^+))\}$

## CONVERSION A AUTOMATA FINITO DETERMINISTICO DEFINICION FORMAL UNA VEZ CONVERTIDO

**NO PUEDE REALIZARSE LA CONVERSIÓN DEBIDO A QUE AFECTA EN NUESTRA EXPRESION REGULAR.**

## EJERCICIO 2:

1. Genera un autómata finito no determinístico que reconozca los nombres de variables, constantes y clases como normalmente los comprendemos en lenguaje de programación.
2. Genera la definición formal, expresión regular, lenguaje por comprensión.
3. Convierte el diagrama a un autómata finito determinístico y genera la definición formal.

### - Definición Formal:

$$AFN < S, \Sigma, p, i, F >$$

Donde:

- **S** = Conjunto finito de estados
- **$\Sigma$**  = Alfabeto de la maquina
- **P** = Subconjunto de  $s^* \Sigma$  llamado relación de Transición
- **i** = Un elemento de S llamado estado inicial
- **F** = Subconjunto de S llamados estados de aceptación

$$S = \{q_0, q_1, q_2\}$$

$$\Sigma = \{\text{letraM}, \text{letram}, \text{numero}, \text{guionBajo}\}$$

$$\text{letraM} = \{[A - Z]\}$$

$$\text{letram} = \{[a - z]\}$$

$$\text{numero} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{guionBajo} = \{\_ \}$$

$$i = \{q_0\}$$

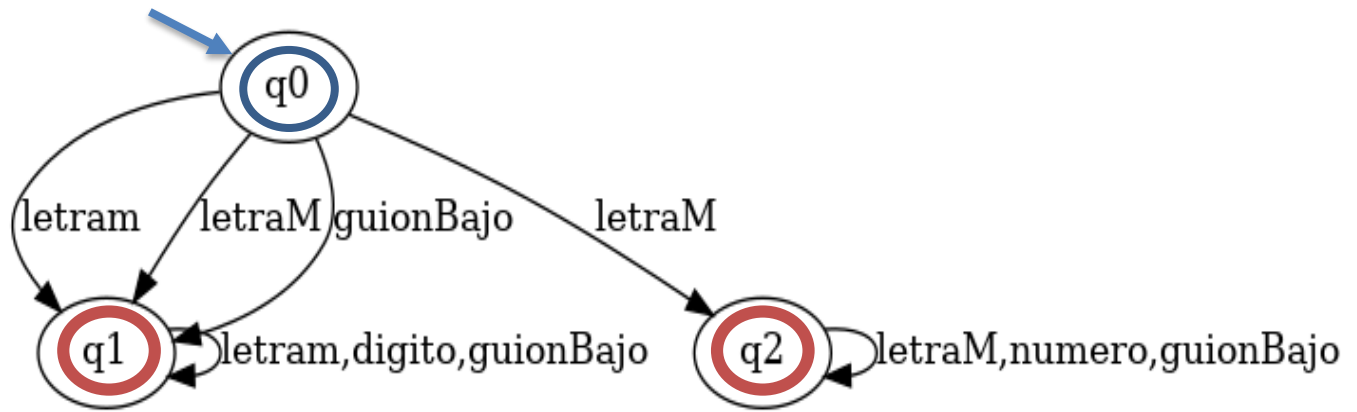
$$F = \{q_1, q_2\}$$

P: Puede estar representada de dos maneras:

$$P = \{((q_0 \text{ letram } q_1), (q_0 \text{ letraM } q_1), (q_0 \text{ letraM } q_2), (q_0 \text{ guionBajo } q_1), (q_1 \text{ numero } q_1), (q_1 \text{ letram } q_1), (q_1 \text{ guionBajo } q_1), (q_2 \text{ numero } q_2), (q_2 \text{ letraM } q_2), (q_2 \text{ guionBajo } q_2))\}$$

EJERCICIO 2					
TABLA DE TRANSICIONES					
		numero	letram	letraM	guion_bajo
inicial	q0		q1	q1,q2	q1
aceptacion	q1	q1	q1		q1
aceptacion	q2	q2		q2	q2
Total Transiciones					10

- Diagrama:

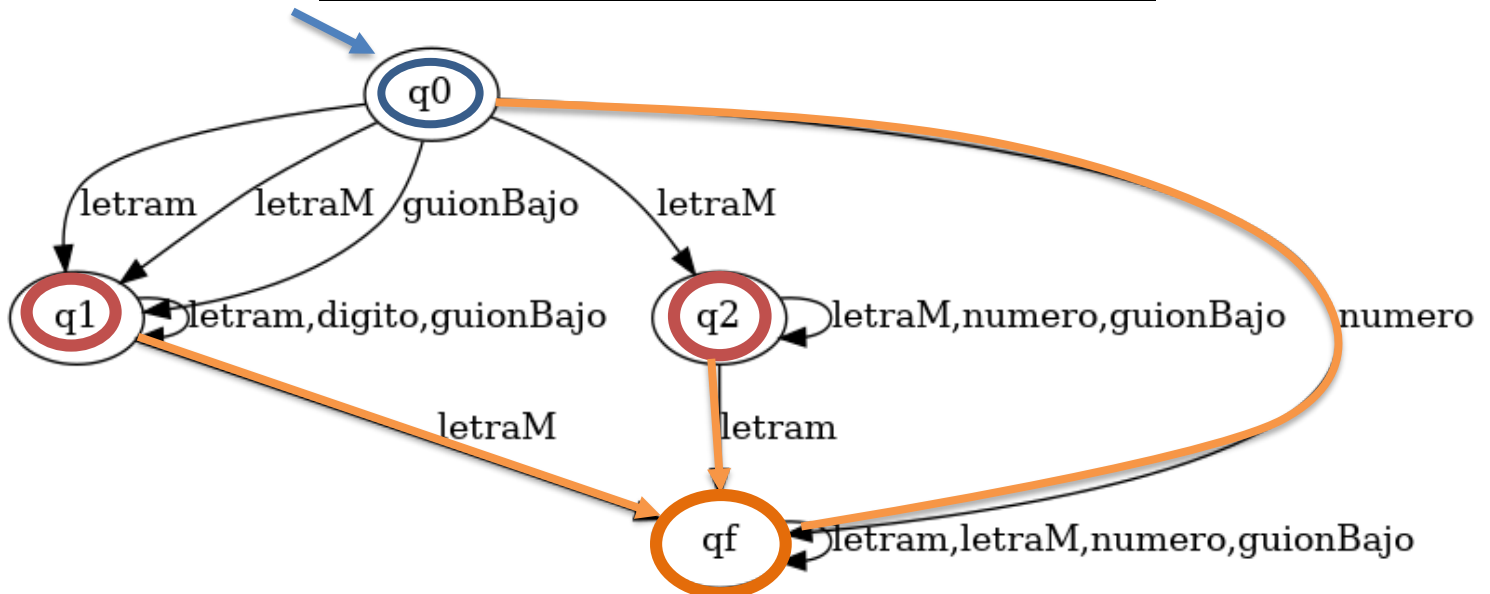


- Lenguaje por Comprensión:

$L =$

$\{w \in \{\text{letraM}, \text{numero}, \text{guiBajo}, \text{letram}\}^* \mid w \text{ cumple con } ((\text{letram}^+ | \text{letraM} | \text{guionBajo}^+) \text{letram}^* | \text{numero}^* | \text{guionBajo}^*) | (\text{letraM}^+ \text{numero}^* \text{guionBajo}^*)\}$

CONVERSION A AUTOMATA FINITO DETERMINISTICO



## DEFINICION FORMAL UNA VEZ CONVERTIDO

### - Definición Formal:

$$AFD < S, \Sigma, \delta, i, F >$$

Donde:

- **S** = Conjunto finito de estados
- **$\Sigma$**  = Alfabeto de la maquina
- **$\delta$**  = Función de transición de  $s^* \Sigma$
- **i** = Estado inicial
- **F** = Conjunto de estados de aceptación

$$S = \{q_0, q_1, q_2, q_3\}$$

$$\Sigma = \{\text{letraM, letram, numero, guionBajo}\}$$

$$\text{letraM} = \{[A - Z]\}$$

$$\text{letram} = \{[a - z]\}$$

$$\text{numero} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\text{guionBajo} = \{\_ \}$$

$$i = \{q_0\}$$

$$F = \{q_2, q_1\}$$

$$P = \{(q_0 \text{ letram } q_1), (q_0 \text{ letraM } q_1), (q_0 \text{ letraM } q_2), (q_0 \text{ guionBajo } q_1), (q_1 \text{ numero } q_1), (q_1 \text{ letram } q_1), (q_1 \text{ guionBajo } q_1), (q_2 \text{ numero } q_2), (q_2 \text{ letraM } q_2), (q_2 \text{ guionBajo } q_2), (q_2 \text{ letram } q_3), (q_0 \text{ numero } q_3), (q_1 \text{ letraM } q_3), (q_3 \text{ letram, letraM, guionBajo, numero } q_3)\}$$

O:

EJERCICIO 2						
TABLA DE TRANSICIONES						
		numero	letram	letraM	guion_bajo	
inicial	q0	q3	q1	q1,q2	q1	5
aceptacion	q1	q1	q1	q3	q1	4
aceptacion	q2	q2	q3	q2	q2	4
	qf	q3	q3	q3	q3	4
Total Transiciones						17

## CONCLUSIONES:

La Elaboración de esta práctica se me hizo un poco laboriosa debido a que me empezaron a salir algunas dudas acerca de los diagramas debido a su construcción se me dificultó crear el primer ejercicio debido a que tenemos que analizar correctamente que las operaciones que pudieran cumplir a cómo lo pidió la docente al final del día logramos elaborar el diagrama para posteriormente pasarlo a computadora usando el programa que llevamos trabajando con anterioridad DrawGraph después de dibujarlo procedemos a hacer su definición formal donde el único momento que se nos dificultó fue la elaboración de la expresión Regular o lenguaje de comprensión debido a que intentamos abarcar las posibles rutas dentro de todo nuestro diagrama para cumplir con el objetivo que era generar operaciones aritméticas y tanto de números enteros como decimales nos costó un poco elaborarlo pero esperemos que esté bien en el apartado de la conversión intentamos realizarla pero nos dimos cuenta que modificaban nuestra expresión regular debido a que se repetían factores y no dejaban trabajar la expresión como debería por ende no pudimos elaborar la primera conversión, ya en el segundo ejercicio simplificamos un diagrama que teníamos anteriormente con 17 estados pero después de un día completo de análisis para ver si funcionaba pudimos reducirlo a solo tres estados esperando que cumplan con lo solicitado por el docente que era generar palabras que admitieran variables funciones y constantes ese fue un poquito más sencillo a elaborar su definición formal de igual manera generar su diagrama y su tabla de transiciones, a diferencia del primero con este sí pudimos trabajar la conversión y la mostramos como una segunda imagen en el PDF.

Esperamos haber cumplido con lo solicitado por el docente y de igual manera esperamos seguir trabajando con más tiempo en plaza estos diagramas y su conversión para adentrarnos más a programar y realizar las prácticas con el objetivo solicitado que es generar un compilador.