

## Trabajo Encargado Nro. 02

### 1.- Objetivo.

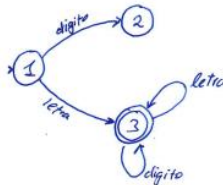
Familiarizarse con la programación de autómatas finitos deterministas utilizando los diagramas de transición o tablas de transición.

### 2.- Desarrollo.

Un identificador puede ser definido por un diagrama de transición y/o tabla de transición, así entonces, podemos presentar soluciones (pseudocódigos) para el reconocimiento de un identificador como se muestra en la figura 01.

Ambas soluciones pueden ser automatizadas, teniendo como dato de entrada: un diagrama, una tabla de transiciones y/o alguna otra estructura de datos de representación.

#### Solucion 1: Mediante diagramas de transición



Variables  
- *Estado*  
- *símbolo*

```
BEGIN
  Estado:= 1;
  símbolo := primer símbolo de la cadena de entrada
  WHILE símbolo no es fin de cadena DO
    CASE Estado OF
      1: IF símbolo es un digito THEN Estado:= 2
          ELSE IF símbolo es una letra THEN Estado:=3
          ELSE salir a rutina de error
      2: Salir a rutina de error
      3: IF símbolo es letra THEN Estado := 3
          ELSE IF símbolo es digito THEN Estado:=3
          ELSE salir a rutina de error
    ENDCASE
    Leer siguiente símbolo de la cadena
  END WHILE
  IF Estado <> 3 THEN
    Salir a rutina de error
  END
```

#### Solucion 2: Mediante Tablas de transición

	digito	letra	FDC
1	2	3	'error'
2	'error'	'error'	'error'
3	3	3	'aceptar'

Donde:

**Tabla[Estado,Entrada]** es el estado actual proveniente del estado anterior leyendo la entrada.

```
BEGIN
  Estado:=1
  REPEAT
    Leer siguiente símbolo de la cadena de entrada
    CASE símbolo OF
      Digito: Entrada:= 'digito'
      Letra : Entrada:= 'letra'
      Fin de cadena : Entrada := 'FDC'
      Otro símbolo : Salir a rutina de error.
    ENDCASE
    Estado:= Tabla[Estado, Entrada]
    IF Estado='error'
      Salir a rutina de error
    UNTIL Estado = 'aceptar'
  END
```

Figura 01. Pseudocódigos para el reconocimiento de un identificador

### 3.- Ejercicios.

- Defina la estructura de datos de representación del diagrama de transición.
- Realizar un generador de código a partir de la estructura definida en el punta 3a.
  - Entrada.  
Archivo en formato gráfico o texto con la representación del autómata finito determinista.
  - Salida  
Código compilable en lenguaje C o C++ para el reconocimiento de una entrada.
- Presentar un banco de pruebas de tres o mas autómatas finitos deterministas para el reconocimiento de un patrón (identificador, numero, etc.)
- Presentar ejemplos de reconocimientos de cadenas a partir de la compilación del archivo de salida del ejercicio 3b y 3c.

#### 4.- Presentación

La tarea debe ser presentada en un documento en formato PDF. Así mismo, el informe debe contener: código fuente, diagramas de transición (AFD), archivos de texto plano (estructuras de datos, banco de pruebas, etc.), capturas de pantalla del funcionamiento de los programas (3b y 3d), explicación del trabajo realizado y conclusiones. De darse el caso, también debe incluir información complementaria que sea pertinente.