

Universidad Tecnológica de Torreon



# Universidad Tecnológica de Torreón

---

## PRÁCTICA DE AUTÓMATAS FINITOS DETERMINISTAS

### **Materia**

Estructuras de Datos Aplicadas

**Erick Israel Rangel Vázquez**  
**Jesús Omar Morales Valenzuela**

### **Asesor**

M.C. Burciaga Pérez Jesús Adrián

Torreón - México

Octubre de 2024

Requisitos de la Cadena:

Se compone de dos secciones, una de dígitos y una de letras mayúsculas, separadas por un \$ o por un & que puede no existir.

La sección de dígitos debe empezar con número par (el cero se considera par) y debe terminar con un número impar. Dicha sección debe medir al menos 4 dígitos en total.

La sección de letras empieza con vocal y deberá terminar con consonante y deberá de medir al menos dos letras de longitud en total (al menos las obligatorias).

Expresion regular

**Digito-par Digits Digits+ Dimpar (\$|&)? vocales letras\* consonante**

Diagrama de Automata

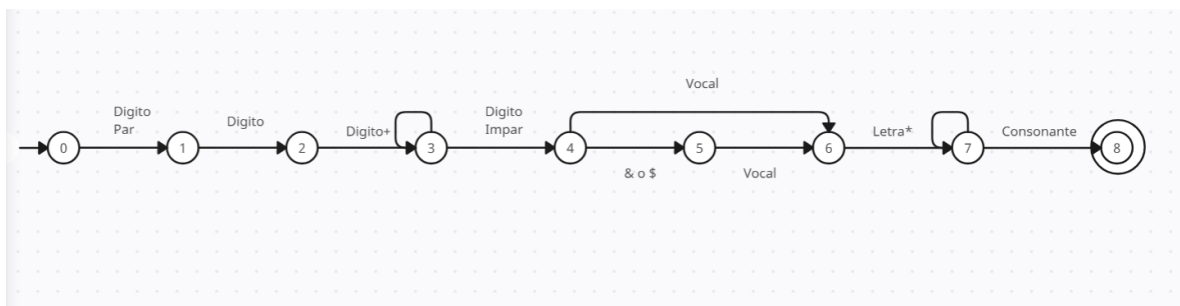


Tabla de Transicion de Estados:

|    | digito | digito par | digito impar | vocal | letra | consonante | \$ | & |
|----|--------|------------|--------------|-------|-------|------------|----|---|
| 0  | 0 1    | 1          | 0            | 0     | 0     | 0          | 0  | 0 |
| 1  | 2      | 2          | 2            | 1     | 1     | 1          | 1  | 1 |
| 2  | 3      | 3          | 3            | 2     | 2     | 2          | 2  | 2 |
| 3  | 3 4    | 3          | 4            | 3     | 3     | 3          | 3  | 3 |
| 4  | 4      | 4          | 4            | 6     | 4 5   | 4          | 5  | 5 |
| 5  | 5      | 5          | 5            | 6     | 5 6   | 5          | 5  | 5 |
| 6  | 6      | 6          | 6            | 7     | 7     | 7          | 6  | 6 |
| 7  | 7      | 7          | 7            | 7     | 7 8   | 8          | 7  | 7 |
| 8* | -      | -          | -            | -     | -     | -          | -  | - |

## Codigo de Programa

```
using System;
using System.Linq;

namespace AutomataApp
{
    public class Automata
    {
        public bool EvaluarCadena(string cadena)
        {
            if (string.IsNullOrEmpty(cadena))
            {
                return false;
            }

            int separatorIndex = -1;
            for (int i = 0; i < cadena.Length; i++)
            {
                if (cadena[i] == '$' || cadena[i] == '&')
                {
                    separatorIndex = i;
                    break;
                }
            }

            string numeros;
            string letras;

            if (separatorIndex != -1)
            {
                numeros = cadena.Substring(0, separatorIndex);
                letras = cadena.Substring(separatorIndex + 1);
            }
            else
            {
                int firstLetterIndex = -1;

                for (int i = 0; i < cadena.Length; i++)
                {
                    if (!EsDigito(cadena[i]))
                    {
                        firstLetterIndex = i;
                        break;
                    }
                }

                if (firstLetterIndex == -1 || firstLetterIndex < 4)
                {
                    return false;
                }

                numeros = cadena.Substring(0, firstLetterIndex);
                letras = cadena.Substring(firstLetterIndex);
            }

            if (!ValidarSeccionNumerica(numeros))
            {
                return false;
            }

            if (!ValidarSeccionLetras(letras))
            {
                return false;
            }

            return true;
        }

        private bool ValidarSeccionNumerica(string numeros)
        {
            if (numeros.Length < 4)
            {
                return false;
            }
        }
    }
}
```

```

        if (!EsDigitoPar(numeros[0]))
        {
            return false;
        }

        if (!EsDigitoImpar(numeros[numeros.Length - 1]))
        {
            return false;
        }

        foreach (char c in numeros)
        {
            if (!EsDigito(c))
            {
                return false;
            }
        }

        return true;
    }

    private bool ValidarSeccionLetras(string letras)
    {
        if (string.IsNullOrEmpty(letras) || letras.Length < 2)
        {
            return false;
        }

        if (!EsVocal(letras[0]))
        {
            return false;
        }

        char ultimoCaracter = letras[letras.Length - 1];
        if (!EsConsonante(ultimoCaracter))
        {
            return false;
        }

        foreach (char c in letras)
        {
            if (!EsLetraMayuscula(c))
            {
                return false;
            }
        }

        return true;
    }

    private bool EsDigito(char c)
    {
        return (c >= '0' && c <= '9');
    }

    private bool EsDigitoPar(char c)
    {
        return (c == '0' || c == '2' || c == '4' || c == '6' || c == '8');
    }

    private bool EsDigitoImpar(char c)
    {
        return (c == '1' || c == '3' || c == '5' || c == '7' || c == '9');
    }

    private bool EsVocal(char c)
    {
        return (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
    }

    private bool EsConsonante(char c)
    {
        return (c >= 'A' && c <= 'Z' && !EsVocal(c));
    }

    private bool EsLetraMayuscula(char c)
    {
        return (c >= 'A' && c <= 'Z');
    }

```

```

}

public class AutomataMenu
{
    private bool continuar = true;
    private Automata automata = new Automata();

    public void MostrarMenu()
    {
        while (continuar)
        {
            Console.Clear();
            Console.WriteLine("Menú:");
            Console.WriteLine("1. Capturar cadena de entrada.");
            Console.WriteLine("2. Créditos.");
            Console.WriteLine("3. Salir.");
            Console.Write("Seleccione una opción: ");

            string opcion = Console.ReadLine();

            switch (opcion)
            {
                case "1":
                    CapturarCadena();
                    break;
                case "2":
                    MostrarCreditos();
                    break;
                case "3":
                    Salir();
                    break;
                default:
                    Console.WriteLine("Opción no válida, intente nuevamente.");
                    break;
            }
        }
    }

    private void CapturarCadena()
    {
        Console.Write("Ingrese la cadena a evaluar: ");
        string cadena = Console.ReadLine();

        if (automata.EvaluarCadena(cadena))
        {
            Console.WriteLine("Cadena válida.");
        }
        else
        {
            Console.WriteLine("Cadena no válida.");
        }

        Console.WriteLine("Presione cualquier tecla para volver al menú...");
        Console.ReadKey();
    }

    private void MostrarCreditos()
    {
        Console.WriteLine();
        Console.WriteLine("Créditos:");
        Console.WriteLine();
        Console.WriteLine("Nombre: Omar Morales Valenzuela");
        Console.WriteLine("Matrícula: 23170090");
        Console.WriteLine("Materia: Estructuras de Datos");
        Console.WriteLine("Grado y sección: 4to Grado, Sección B");
        Console.WriteLine();
        Console.WriteLine("Nombre: Erick Israel Rangel Vazquez");
        Console.WriteLine("Matrícula: 23170090");
        Console.WriteLine("Materia: Estructuras de Datos");
        Console.WriteLine("Grado y sección: 4to Grado, Sección B");

        Console.WriteLine("Presione cualquier tecla para volver al menú...");
        Console.ReadKey();
    }

    private void Salir()
    {
        continuar = false;
    }
}

```

```
        Console.WriteLine("Saliendo del programa...");
    }
}

class Program
{
    static void Main(string[] args)
    {
        AutomataMenu menu = new AutomataMenu();
        menu.MostrarMenu();
    }
}
```