

## Manual técnico

El proyecto fue realizado en el lenguaje de programación C#, plataforma .NET 5.0, la herramienta utilizada para el desarrollo fue Visual Studio 2019.

### Clases desarrolladas:

#### Class Estado:

Atributos:

- `_nombre`: de tipo string, almacena el nombre del estado.
- `_esInicial`: de tipo bool, almacenará si el estado es inicial o no.
- `_esDeAceptacion`: de tipo bool, almacenará si el estado es de aceptación o no.
- `_transiciones`: de tipo Dictionary, almacenará las transiciones del estado.

#### Class AF:

Representa un autómata finito determinístico o no determinístico.

Atributos:

- `_simbEntrada`: de tipo List, almacena los símbolos de entrada del autómata finito.
- `_estados`: de tipo Dictionary, almacena los estados del autómata finito.
- `_estIniciales`: de tipo List, almacena los estados iniciales del autómata finito.
- `_estDeAceptacion`: de tipo List, almacena los estados de aceptación del autómata finito.

Métodos:

- `agregarTransiciones`: recibe como parámetro una matriz que contiene las transiciones y las agrega al autómata finito.
- `toAFD`: retorna el actual autómata finito convertido en autómata finito determinístico.
- `toString`: retorna un string con los símbolos de entrada, estados, estados iniciales y estados de aceptación del autómata finito.

#### Class AFD:

Representa un autómata finito determinístico.

Atributos:

- `_simbEntrada`: de tipo List, almacena los símbolos de entrada del autómata finito determinístico.

- `_estados`: de tipo Dictionary, almacena los estados del autómata finito determinístico.
- `_estInicial`: de tipo string, almacena el estado inicial del autómata finito determinístico.
- `_estDeAceptacion`: de tipo List, almacena los estados de aceptación del autómata finito determinístico.

#### Métodos:

- `agregarTransiciones`: recibe como parámetro una lista con las transiciones y las agrega al autómata finito determinístico.
- `simplificar`: simplifica el autómata finito determinístico llevándolo a su forma mínima.
- `toString`: retorna un string con los símbolos de entrada, estados, estados iniciales y estados de aceptación del autómata finito determinístico.
- `evaluar`: recibe como parámetro una lista con los símbolos de la cadena a evaluar y retorna una lista con los estados que se recorren.

#### Class Form1:

Derivada de la clase Form es una clase con interface gráfica de usuario, lo que le permitirá al usuario interactuar con el programa.

#### Atributos:

- `_autFinito`: de tipo AF, almacenará un autómata finito.
- `_autFinDet`: de tipo AFD, almacenará un autómata finito determinístico.
- `_tblIngrTransiciones`: de tipo TableLayoutPanel, nos permitirá mostrar una tabla para ingresar las transiciones del autómata.
- `_tblAutomataMinimo`: de tipo TableLayoutPanel, nos permitirá mostrar una tabla con el autómata finito determinístico mínimo.
- `_tblPasoPaso`: de tipo TableLayoutPanel nos permitirá mostrar el reconocimiento paso a paso de la cadena que se vaya a evaluar.
- `_tb_tblIngrTransiciones`: de tipo TextBox[,], no ayudará a guardar una matriz los TextBox con las transiciones del autómata finito ingresado.
- `_recorrido`: de tipo List, almacenará los estados que se recorren al evaluar una cadena.
- `_entradas`: de tipo List, almacenará los símbolos de la cadena que se desee evaluar.
- `_anterior`: de tipo Estado, nos será de ayuda para mostrar el recorrido.
- `_actual`: de tipo Estado, nos será de ayuda para mostrar el recorrido.

- `_simbolo`; de tipo string, nos será de ayuda para mostrar el recorrido.
- `_n`; de tipo int, nos será de ayuda para mostrar el recorrido.

#### Métodos:

- `_KeyPress`: es un evento que se ejecuta cada vez que se intenta modificar el texto de un TextBox, nos ayudará a evitar que se ingresen símbolos diferentes a números, letras y comas.
- `btn_CrearAutomata_Click`: es un evento que se ejecuta al dar click sobre el botón crear, crea un nuevo autómata finito.
- `crearTblIngrTransiciones`: crea la tabla que nos permite ingresar las transiciones del autómata finito.
- `btn_ValidarAutomata_Click`: es un evento que se ejecuta al dar click sobre el botón validar, valida que las transiciones se hayan ingresado correctamente, ejecuta el método que agrega esas transiciones al autómata finito y lo convierte a determinístico, ejecuta el método que muestra el autómata finito determinístico.
- `crerTblAutomataMinimo`: crea una tabla y muestra el autómata finito determinístico.
- `btn_EvaluarCadena_Click`: es un evento que se ejecuta al dar click sobre el botón evaluar, verifica que la cadena ingresada pueda ser reconocida por el autómata, evalúa la cadena y muestra un mensaje informando si la cadena es de aceptación o de rechazo, ejecuta el método que crea una tabla con el autómata para mostrar el reconocimiento paso a paso.
- `btn_EditTransiciones_Click`: nos permite volver a editar las transiciones después de haber validado el autómata.
- `btn_NuevoAutomata_Click`: nos permite crear un nuevo autómata.
- `btn_EditAutomata_Click`: nos permite editar los estados de un autómata después de que este ya ha sido creado.
- `btn_SimpAutomata_Click`: simplifica el autómata, y actualiza la tabla para mostrar el nuevo autómata mínimo.
- `btn_SiguientePaso_Click`: nos permite ir viendo paso a paso las transiciones al ir evaluando los símbolos de una cadena, ejecuta el método que actualiza la tabla para ver la transición.
- `crearTblPasoPaso`: crea una tabla con el autómata para mostrar el reconocimiento paso a paso.
- `actualizarTablaPasoApaso`: nos permitirá mostrar la transición al procesar un símbolo de la cadena.
- `ultimaEntrada`: mostrara un mensaje indicando si la cadena es de aceptación o de rechazo.