



26 DE SEPTIEMBRE DE 2020

# MANUAL DE USUARIO

KEVIN JOSUÉ CALDERÓN PERAZA

201902714



## DESCRIPCION GENERAL DEL PROGRAMA

El tema de autómatas es un tema muy importante para el curso lenguajes formales y programación, en este caso el programa en consola, para que el estudiante pueda aprender como aprender a programar y a conocer más sobre el tema sobre las autómatas deterministas y gramáticas regulares.

En este, es se puede agregar AFD de forma manual ingresando uno a uno cada parte del autómata, así como una carga masiva de datos por medio de un archivo. También para que el usuario pueda ingresar visualizar el autómata y la gramática de forma gráfica, se encuentra un apartado de reportes, para tener una mejor visualización de los autómatas y las gramáticas. Otro de las funcionalidades es para que casos es válida ese autómata pudiendo validar cadenas y que ruta es la más fácil para la solución de la misma.

## REQUERIMIENTOS MINIMOS DEL SISTEMA

Los requerimientos son los que se utilizaron para la creación del proyecto y los anteriores. Puede funcionar con sistemas operativos de menor versión sin ningún problema.

- CPU: Intel Core i3-4030 CPU @1.90Ghz
- RAM: 8GB
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64.
- S.O: Windows 10 PRO, versión 1909.

## DESCRIPCION DE LOS PASOS PARA EL USO DE LAS OPCIONES MAS IMPORTANTES

### MENU PRINCIPAL

En este menú principal se encuentran los diferentes módulos para que el usuario pueda escoger y acceder a las diferentes funcionalidades. También se encuentra el modulo “acerca de...” para conocer los catedráticos y a que clase pertenece este proyecto.

```
*****
MENU PRINCIPAL
*****
1.Modulo AFD
2.Modulo de gramaticas regulares
3.Acerca de...
0.Salir
Elige una opcion:
def mostrarMenu():
    print("*****")
    print("LISTO")
    print("*****")
```

## MODULO AFD

En este módulo se encuentra todo lo relacionado con los autómatas finitos deterministas. En ella se puede realizar diferentes funcionalidades desde agregar manualmente hasta generar reportes.

```
*****
MODULO AFD
*****
1.Crea AFD
2.Cargar archivo
3.Evaluar cadena
4.Guardar AFD en archivo
5.Generar reporte AFD
6.Generar gramatica regular
0.Regresar al menu principal

Elige una opcion:
```

```
DESCRIPCION DE LOS
MAS IMPORTANTES

*****
MENU PRINCIPAL
*****
1.Modulo AFD
2.Modulo de gramaticas reg
3.Acerca de...
0.Salir

Elige una opcion:
```

Crear Afd: La funcionalidad de que hace esta opción es crear un autómata de forma manual agregando sus estados y alfabetos, siempre tomando en cuenta todas las restricciones que puede tener un autómata y sus reglas.

```
Escribe el nombre de la automata: demo1

Escribe 'fin' para terminar de escribir los estados, alfabetos y aceptacion.
Escribe los estados: x1
Escribe los estados: x2
Escribe los estados: fin
Ingrese los alfabetos:
Ingrese los alfabetos: 0
Ingrese los alfabetos: 1
Ingrese los alfabetos: fin
Escribe el estado inicial: x1
Escribe los estados de aceptacion: x2
Escribe los estados de aceptacion: fin
Escribe tu transicion: _
```

```
*****
MODULO AFD
*****
1.Cargar archivo
3.Evaluar cadena
4.Guardar AFD en archivo
5.Generar reporte AFD
6.Generar gramatica regular
0.Regresar al menu principal

Elige una opcion:
```

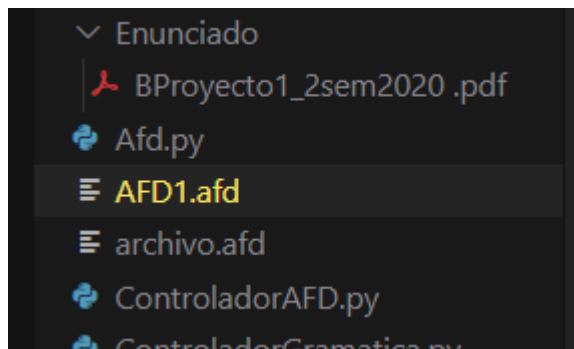
Cargar archivo: La opción dos, carga un archivo con extensión. afd estos archivos se lee línea por línea para luego ser guardado en la memoria dinámica.

```
*****
MODULO AFD
*****
1.Crea AFD
2.Cargar archivo
3.Evaluar cadena
4.Guardar AFD en archivo
5.Generar reporte AFD
6.Generar gramatica regular
0.Regresar al menu principal

Elige una opcion: 2
Escribe el nombre del archivo: archivo.afd_
```

### Guardar AFD en archivo

Para guardar el autómata en un archivo se tiene que escribir el nombre del autómata que anteriormente fue ingresada, ya sea manualmente o por medio del archivo. Si es el caso, que no se encuentra el autómata escrita te muestra un mensaje de advertencia. El archivo es creado en el mismo directorio donde se encuentra tu proyecto.



```
Elige una opcion: 4
Escribe el nombre de la Afd para generar el archivo: AFD1

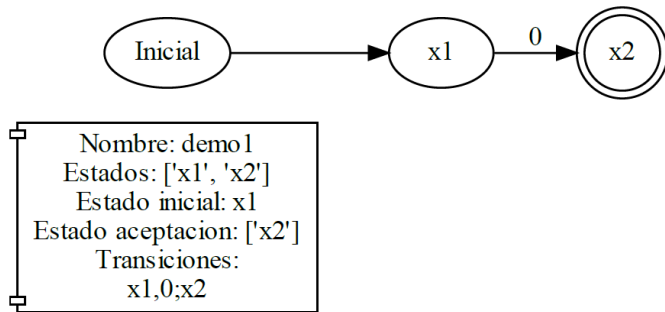
Automata encontrado y archivo generado con exito :D

*****
MODULO AFD
*****
1.Crea AFD
2.Cargar archivo
3.Evaluar cadena
4.Guardar AFD en archivo
5.Generar reporte AFD
6.Generar gramatica regular
0.Regresar al menu principal

Elige una opcion: 4
Escribe el nombre de la Afd para generar el archivo: asdf

La automata ingresada no se encuentra en la lista :D
```

Generar reporte AFD: En esta funcionalidad se muestra una lista de afd's ingresados anteriormente y tienes que ingresar una opción para el reporte que quieras generar. El reporte muestra un grafo de cómo es la forma que se representa gráficamente el autómata con sus relaciones. Y el reporte se genera en el directorio donde tienes el código fuente del proyecto



```

*****
      MODULO AFD
*****
1.Crea AFD
2.Cargar archivo
3.Evaluar cadena
4.Guardar AFD en archivo
5.Generar reporte AFD
6.Generar gramatica regular
0.Regresar al menu principal

Elige una opcion: 5
*****
      LISTA DE AFD'S
*****
0.demo1
1.AFD1
2.AFD2
3.AFD3

Selecciona la automata para el reporte: 0
_
Reporte generado con exito :D
  
```

Generar gramática regular: Esta funcionalidad muestra la AFD como es que se representa de forma de gramática regular, igualmente funciona seleccionando una afd anteriormente ingresada y aparece en consola el resultado final, también pregunta si lo quieres guardar en un archivo o no.

```

Elige una opcion: 6
*****
      LISTA DE AFD'S
*****
0.demo1
1.AFD1
2.AFD2
3.AFD3

Escribe la automata para la gramatica: 0

-----
Terminales: [' ', '0', '1']
No terminales: ['x1', 'x2']
Inicio: x1
Producciones:
x1>0 x2

Escribe 'si', si quieres generar el archivo o 'no' para continuar...no
  
```

## MODULO GRAMATICA REGULAR

En este módulo se encuentra todo relacionado con las gramáticas regulares y tiene funcionalidades parecida que el modulo anterior. Siempre tomando en cuenta que se puede navegar entre opciones seleccionando entre números.

```
*****
MODELO GRAMATICA REGULAR
*****
1.Crea gramatica regular
2.Cargar archivo de entrada
3.Evaluar cadenas
4.Eliminar recursividad por la izquierda
5.Guardar gramatica en archivo
6.Generar reporte gramatica regular
0.Regresar al menu principal

Elige una opcion:
```

Crear gramática regular: Para esta funcionalidad es parecida al módulo anterior, tomando en cuenta las consideraciones de falta de no terminales al ingresar las producciones, etc. Igualmente, ingresadas uno por uno cada terminal, no terminal y las producciones.

```
*****
MODELO GRAMATICA REGULAR
*****
1.Crea gramatica regular
2.Cargar archivo de entrada
3.Evaluar cadenas
4.Eliminar recursividad por la izquierda
5.Guardar gramatica en archivo
6.Generar reporte gramatica regular
0.Regresar al menu principal

Elige una opcion: 1
Escribe el nombre de la gramatica: demo1
Escribe un no terminal: a
Escribe un no terminal: b para la gramatica: 0
Escribe un no terminal: c
Escribe un no terminal: fin
Escribe un terminal: 0
Escribe un terminal: 1
Escribe un terminal: fin
Escribe un no terminal Inicial: a
Escribe un no terminal Inicial: fin
Escribe las producciones: a>1 b
Escribe las producciones: b>0 c
Escribe las producciones: fin
Gramatica agregada con exito! :D
```

\*Nota, las demás funcionalidades del módulo dos. Cumplen con las mismas condiciones y la misma forma de ingresar datos que el modulo anterior.

## **MODULO ACERCA DE**

En este módulo muestra en consola información del curso, catedráticos y auxiliares que imparten la clase y la sección a la que fue asignada para el segundo semestre en el año 2020.

```
*****
LENGUAJES FORMALES Y DE PROGRAMACION
Catedratica: Inga. Zulma Karina Aguirre Ordonez
Aux: Danilo Urias Coc
Seccion: B
*****
```

## **GLOSARIO DE TERMINOS**

- Cadena vacía: En el caso del Proyecto, se representa con este símbolo \$ y representa un estado de aceptación, quiere decir que en ese estado puede terminar la gramática.
- Edge: En el framework de graphviz y crear archivos, es un método que sirve para unir nodos, ya sea creados manualmente o creados por ciclos.
- Grafo: Representación simbólica de los elementos constituidos de un sistema o conjunto, mediante esquemas gráficos.
- Node: En el framework de graphviz, es el método que sirve para crear nodos e imprimirlos en un archivo, ya sea pdf o una imagen.
- Proceso de expansión: Es la forma de representar autómatas o gramáticas de forma expandida para una mejor visualización de una ruta que lleva una afd.
- Producciones: En las Gramáticas regulares, es el proceso de leer de un no terminal a otro, por medio de los terminales hasta que encuentra un estado de aceptación.
- Transiciones: En las AFD's, es el proceso de pasar de un estado a otro por medio de un alfabeto.

- Write: Un método nativo de Python para escribir dentro de un archivo sin importar la extensión de ese archivo.

## SECCIÓN DE PREGUNTAS FRECUENTES

- **¿Cómo agrego archivos al proyecto?** Escogiendo la opción correcta que se muestra en el menú, y solo agregando el nombre de archivo y su tipo de extensión, recordando que solo acepta archivos con extensión. afd y. gre.
- **¿En qué formato genera los reportes?** Los reportes se generan en el formato pdf.
- **¿Cómo agrego los estados de forma manual?** Luego de ingresar a la opción de crear afd, automáticamente el programa te va lanzando las opciones para que puedas llenarlas sin preocuparte de buscar más opciones.
- **¿En qué ubicación se guarda el archivo generado de mi autómata?** El archivo se encuentra en el mismo directorio donde está el proyecto.
- **¿Las transiciones tienen que tener un orden específico al agregarlas manualmente?** De preferencia es que, si se ingresen en orden, pero al final no importan, solo tienes que tomar en cuenta que los estados y los alfabetos tuviste que haberlos ingresado anteriormente.
- **¿En qué extensión admite archivos el proyecto?** Solo archivos con extensión .afd y .gre