

AP

Automáta de Pila

MANUAL DE USUARIO

Contenido

INTRODUCCIÓN:.....	3
Iniciar Aplicación	4
.....	5
Menú Principal.....	5
1. Cargar archivo	5
2. Mostrar información general de la gramática.....	6
3. Generar autómata de pila equivalente	7
4. Reporte de recorrido	8
5. Reporte en tabla	9
6. Salir.....	11
Estructura y formato de los archivos	12
CONCLUSIÓN:.....	13
RECOMENDACIONES:.....	14

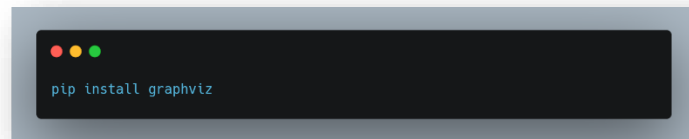
INTRODUCCIÓN:

Este documento es un instructivo de ayuda para todo aquel que haga uso de esta aplicación generador de autómatas de pila y analizador de cadenas, con el fin de demostrar las distintas funcionalidades que ofrece y así obtener el mejor rendimiento de esta aplicación.

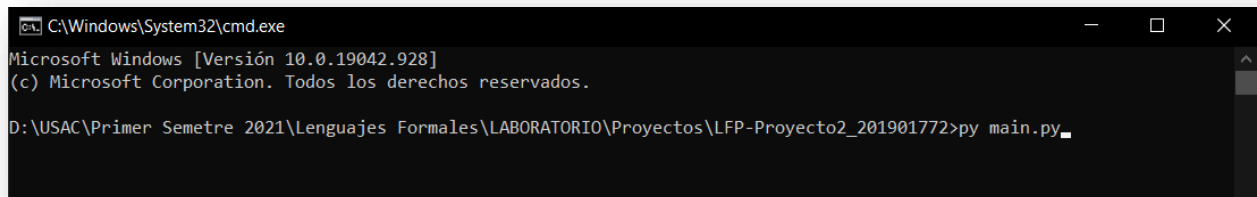
Iniciar Aplicación

Para poder ejecutar la aplicación es necesario tener instalado lo siguiente en su ordenador:

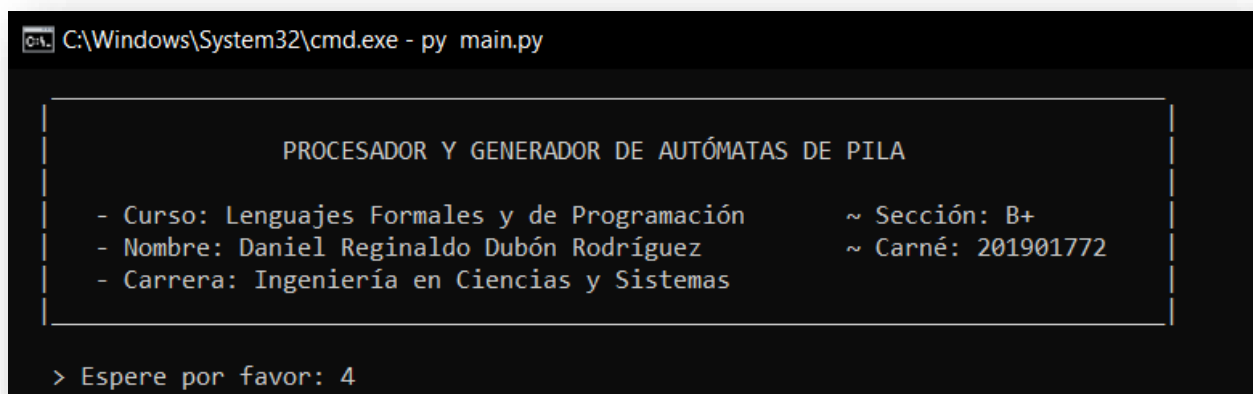
- Python en la versión 3.8.1 en adelante, puede descargarlo si no lo tiene desde su pagina oficial: <https://www.python.org/downloads/>.
- Graphviz puede descargarlo desde su pagina oficial: <https://www.graphviz.org/download/>.
- Paquete de python graphviz, si no se tiene este paquete puede instalarlo ejecutando el siguiente comando desde una terminal.



Ya teniendo instalado los requerimientos proceda a abrir una terminal/consola en el directorio donde se encuentra el archivo `main.py`, y luego ejecute el siguiente comando `python main.py`.



Al ejecutar el comando se iniciará el programa y se le mostrará unas animaciones donde podrá observar la información del desarrollador y luego automáticamente se le mostrará un menú con las siguientes opciones:



```
C:\Windows\System32\cmd.exe - py main.py

----- MENÚ PRINCIPAL -----

1. Cargar archivo
2. Mostrar información general de la gramática
3. Generar autómata de pila equivalente
4. Reporte de recorrido
5. Reporte en tabla
6. Salir

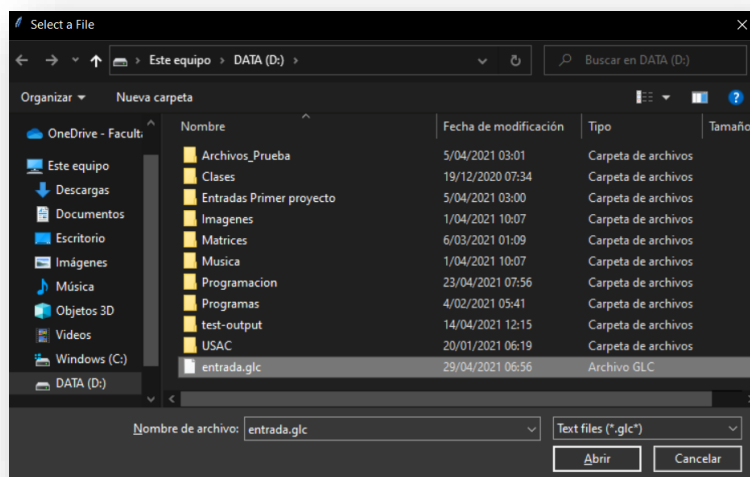
Ingrese una opción:
> █
```

Menú Principal

Esta muestra todas las opciones disponibles de la aplicación.

1. Cargar archivo

Esta opción desplegará un explorador de archivos el cual permitirá buscar el archivo que se desea cargar para posteriormente ser analizado.



Al seleccionar el archivo este se cargará en la aplicación y mostrará un mensaje si fue cargado el archivo correctamente o no.

```
-----CARGAR ARCHIVO-----

> Archivo cargado

- PRESIONE ENTER PARA CONTINUAR...
```

CONSIDERACIONES: La aplicación solo admitirá archivos con la extensión “.glc”.

Posteriormente se analizará el archivo cargado y se generará un reporte en donde se mostrará todas aquellas gramáticas que no sean libre de contexto que no fueron admitidas por el programa.



2. Mostrar información general de la gramática

Esta opción del menú permite mostrar todos los nombres de gramáticas que se encuentran actualmente en el sistema para que el usuario pueda elegir una.

```
C:\Windows\System32\cmd.exe - py main.py

----- SELECCIONAR GRAMÁTICA -----

1. Grm1
2. Grm3
3. Gramatica satanas
4. Doble terminal
5. NT IND
6. CUSTOM
7. Regresar

- Ingrese una opción:
> 
```

Cuando el usuario elija una gramática, inmediatamente se mostrará la información de la gramática en la consola. Los datos que deben mostrarse son los siguientes:

```
C:\Windows\System32\cmd.exe - py main.py

----- INFORMACIÓN DE LA GRAMÁTICA -----

- Nombre de la gramática tipo 2: Grm1
- No Terminales: = { S,A,B,C }
- Terminales = { a,b,z }
- No Terminal Inicial = S
- Producciones:

  S -> A
  A -> a A a
    | B
  B -> b B b
    | C
  C -> z C
    | z

- PRESIONE ENTER PARA CONTINUAR..._
```

3. Generar autómeta de pila equivalente

Esta opción permite generar un autómeta de pila a partir de una gramática independiente del contexto cargadas en el sistema, se le desplegara la lista de gramáticas disponibles para generar el autómeta.

```
C:\Windows\System32\cmd.exe - py main.py

----- SELECCIONAR GRAMÁTICA -----

1. Grm1
2. Grm3
3. Gramatica satanas
4. Doble terminal
5. NT IND
6. CUSTOM
7. Regresar

- Ingrese una opción:
>
```

Al seleccionar una gramática el programa generará un reporte del autómeta de pila generado y también se cargará en el sistema para posterior mente validar cadenas y generar los distintos reportes.

```

C:\Windows\System32\cmd.exe - py main.py

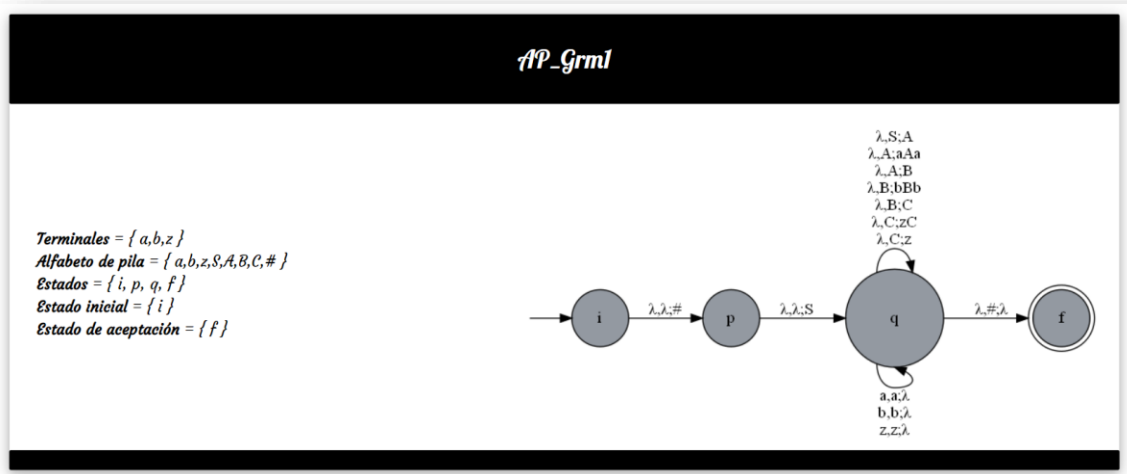
----- GENERAR AUTOMÁTA DE PILA -----

> Automáta de pila generado

- PRESIONE ENTER PARA CONTINUAR..._

```

Reporte generado:



4. Reporte de recorrido

El usuario podrá elegir uno de los autómatas de pila generados en la opción 3.

```

C:\Windows\System32\cmd.exe - py main.py

----- SELECCIONAR AUTOMÁTA DE PILA -----

1. AP_Grm1
2. AP_CUSTOM
3. Regresar

- Ingrese una opción:
> _

```


Finalizada la elección del autómata, se solicitará al usuario el ingreso de una cadena de entrada para que sea validada con el autómata. Esta opción permitirá generar un reporte en html que contendrá a detalle cada una de las iteraciones realizadas para validar la cadena.

```
C:\Windows\System32\cmd.exe - py main.py

----- VERIFICAR CADENA -----

Ingrese la cadena a validar:
> dann_
```

5. Reporte en tabla

El usuario podrá elegir uno de los autómatas de pila generados en la opción 3. Esta opción deberá mostrar como resultado un reporte en html con una tabla de resumen.

```
C:\Windows\System32\cmd.exe - py main.py

----- SELECCIONAR AUTOMÁTA DE PILA -----

1. AP_Grm1
2. AP_CUSTOM
3. Regresar

Ingrese una opción:
> _
```

Finalizada la elección del autómata, se solicitará al usuario el ingreso de una cadena de entrada para validar si se acepta no en el autómata.

```
C:\Windows\System32\cmd.exe - py main.py

----- VERIFICAR CADENA -----

Ingrese la cadena a validar:
> dann_
```

Esta opción deberá mostrar como resultado un reporte en html con una tabla de resumen.

Reporte en Tabla

Cadena: dann

Iteración	Pila ←	Entrada	Transiciones
0		d	$(\{ \lambda, \lambda; p, \# \})$
1	#	d	$(p, \lambda, \lambda; q, S)$
2	S#	d	$(q, \lambda, S; q, C, A)$
3	AC#	d	$(q, \lambda, A; q, B, d)$
4	dB#	d	$(q, d, d; q, \lambda)$
5	BC#	a	$(q, \lambda, B; q, C, a)$
6	aCC#	a	$(q, a, a; q, \lambda)$
7	CC#	n	$(q, \lambda, C; q, n)$
8	nC#	n	$(q, n, n; q, \lambda)$
9	C#	n	$(q, \lambda, C; q, n)$
10	n#	n	$(q, n, n; q, \lambda)$
11	#	λ	$(q, \lambda, \#, f, \lambda)$
12	λ	λ	f

Cadena válida

```
C:\Windows\System32\cmd.exe - py main.py

----- VERIFICAR CADENA -----

Ingrese la cadena a validar:
> dann

> Cadena válida

> Reporte generado

- PRESIONE ENTER PARA CONTINUAR..._
```

6. Salir

Esta opción terminara la ejecución de la aplicación.

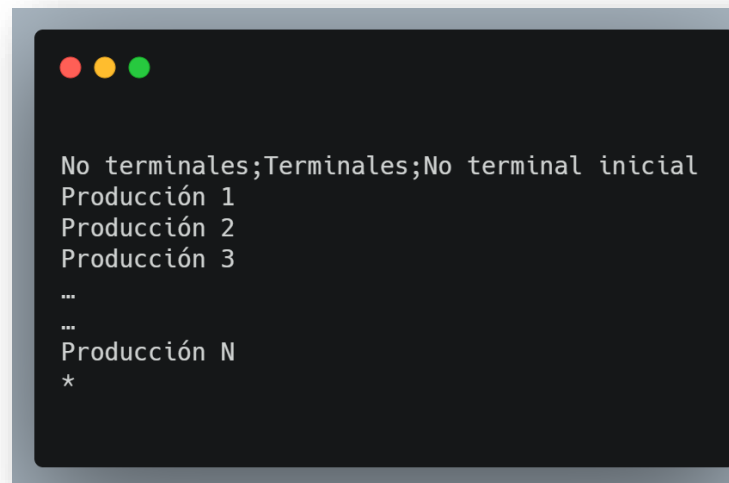
```
- Ingrese una opción:  
> 6  
> Saliendo...
```

Estructura y formato de los archivos

Dentro del archivo de entrada podrán venir N gramáticas, el final de cada gramática estará marcado por un asterisco (*).

Las partes de cada gramática estarán especificadas en el orden siguiente:

Para el caso de los no terminales y terminales vendrán separados por comas.



```
No terminales;Terminales;No terminal inicial
Producción 1
Producción 2
Producción 3
...
...
Producción N
*
```

Ejemplo de archivo de entrada:

		entrada.glc
Nombre	1	Grml
No Terminales;Terminales;Terminal inicial	2	S,A,B,C;a,b,z;S
Producción 1	3	S->A
Producción 2	4	A->a A a
Producción 3	5	A->B
Producción 4	6	B->b B b
Producción 5	7	B->C
...	8	C->z C
Producción n	9	C->z
Fin de la gramática	10	*
Nombre	11	Gramatica2
No Terminales;Terminales;Terminal inicial	12	A,B,C,D;0,1;A
Producción 1	13	A->1 B
Producción 2	14	A->0 C
Producción 3	15	B->1 A
Producción 4	16	B->0 D
Producción 5	17	C->1 D
Producción 6	18	C->0
...	19	D->1 C
...	20	C->0 A
Producción n	21	D->0 B
Fin de la gramática	22	*
	23	
	24	
	...	
	...	

CONCLUSIÓN:

Un autómata de pila es una buena aplicación para procesar información en archivos de texto que permite reconocer gramáticas y a través de ello validar cadenas, los AP también nos es de gran utilidad para realizar un analizador sintáctico que es muy utilizado en los compiladores, solo cumpliendo con la estructura establecida para los archivos de entrada se podrá generar AP en donde se podrá validar cadenas que reconozca la gramática. Con esta aplicación es indispensable generar reportes ya que ayudan a conocer el proceso que se llevo a cabo para validar cada cadena.

RECOMENDACIONES:

- Leer el manual de usuario para obtener el mejor rendimiento de la aplicación.
- Utilizar el formato y estructura establecida para la creación de los archivos de texto, para que la aplicación pueda analizar y procesarla.