

# MANUAL DE USUARIO BASILISK

Eduardo Tomas Ixen Rucuch

LAB. LENGUAJES FORMALES Y DE PROGRAMACION USAC

## INDICE

<b>INDICE</b>	<b>1</b>
<b>INTRODUCCION</b>	<b>2</b>
<b>REQUISITOS PREVIOS</b>	<b>2</b>
<b>EJECUCION DEL PROGRAMA</b>	<b>2</b>
<b>OPCIONES</b>	<b>3</b>
Opcion 1. Cargar script:	3
Opción 2. Manejo AFD:	3
Opción 3. Pila interactiva:	4
Opción 4: Diagrama:	6
Opción 5. Salir:	6

## INTRODUCCION

Basilisk es una herramienta que funciona por medio de un menú que proporciona varias funciones que tienen como objetivo cargar el contenido de un script a memoria para analizar que todos los caracteres pertenezcan al lenguaje definido, genera un reporte para visualizar los caracteres aceptados y los que no lo son, así mismo es capaz de analizar que el texto aparezca en el orden correcto. Por último es posible generar un diagrama para visualizar la secuencia del texto cargado a memoria.

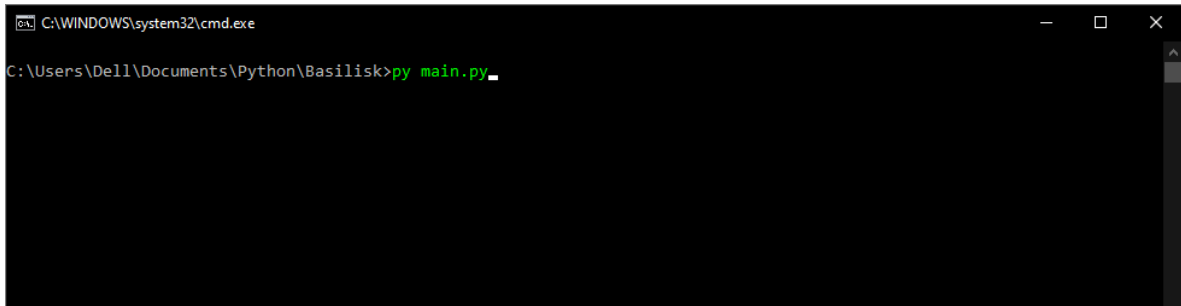
## REQUISITOS PREVIOS

1. Contar con la versión de Python 3.8
2. Contar con Graphviz
3. Instalar la librería Tabulate para Python
4. Conocimientos sobre ECMAScript o ES6 de javascript para crear los archivos de entrada.

## EJECUCION DEL PROGRAMA

Para ejecutar el programa únicamente se necesita acceder al símbolo del sistema (CMD), localizarse dentro de la carpeta del proyecto y teclear el siguiente comando:

```
py main.py
```



Al ejecutar el comando se obtendrá como resultado el siguiente menú:

```
//////////////////////////////// MENU //////////////////////////////////
1. Cargar script
2. Manejo AFD
3. Pila interactiva
4. Diagrama
5. Salir
Ingrese la opcion a seleccionar
|
```

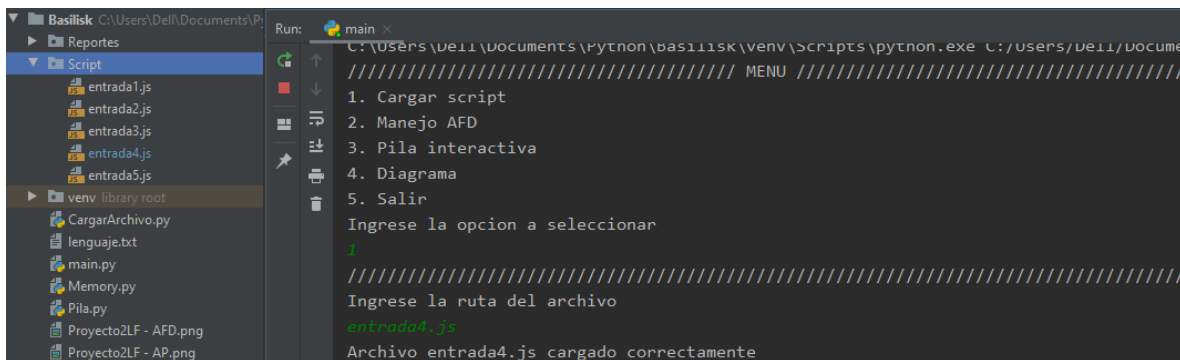
La funcionalidad de cada opción se explica en la sección de OPCIONES.

## OPCIONES

### Opcion 1. Cargar script:

Esta opción permite cargar un archivo con contenido javascript a memoria para ser analizado. Puede ser mediante la ruta completa del archivo o simplemente con su nombre y extensión pero colocándolo previamente en la carpeta “Script” del proyecto.

Si el archivo es cargado correctamente se mostrará un mensaje indicándolo, de lo contrario se muestra un mensaje de error.



The screenshot shows a terminal window with a file explorer on the left. The file explorer displays a directory structure with a 'Script' folder containing five JavaScript files (entrada1.js to entrada5.js) and a 'venv' folder. The terminal window shows the following output:

```
Run: C:\Users\ DELL \Documents \Python \basiliisk \venv \Scripts \python.exe C:\Users\ DELL \Documents \Python \basiliisk \venv \Scripts \python.exe
//////////////////////////////// MENU //////////////////////////////////
1. Cargar script
2. Manejo AFD
3. Pila interactiva
4. Diagrama
5. Salir
Ingrese la opcion a seleccionar
1
////////////////////////////////
Ingrese la ruta del archivo
entrada4.js
Archivo entrada4.js cargado correctamente
```

### Opción 2. Manejo AFD:

Esta opción escanea el contenido del archivo cargado revisando que todas las palabras y caracteres pertenezcan javascript. Al finalizar el escaneo se generan dos reportes en HTML, uno con todas las palabras y caracteres que pertenecen al lenguaje y otro con los que no pertenecen.

El archivo cargado a memoria es el siguiente:

```
let @ variable1 = true;
if (true) + { #
    & print("Archivo de prueba!!");
}
```

Al ejecutar la opción 2 produce como resultado:

```
Archivo entrada4.js cargado correctamente
//////////////////////////////// MENU //////////////////////////////////
1. Cargar script
2. Manejo AFD
3. Pila interactiva
4. Diagrama
5. Salir
Ingrese la opcion a seleccionar
2
Desea abrir el reporte Tokens_Validos? S/N
s
Desea abrir el reporte Errores_Encontrados? S/N
s
```

Tokens_Validos					
No	Nombre	Valor	Descripcion	Linea	Columna
1	tk_let	let	Palabra reservada para declarar una variable	1	1
2	tk_identificador	variable1	Palabra que funciona como identificador	1	7
3	tk_igual	=	Indica asignacion de un valor	1	17
4	tk_true	true	Booleano verdadero	1	19

Errores_Encontrados			
No	Error	Linea	Columna
1	@	1	5
2	+	2	11
3	#	2	15
4	%	2	17
5	&	2	19

Si se ejecuta esta opción y no se ha cargado un archivo a memoria, se muestra un mensaje de error.

### Opción 3. Pila interactiva:

Esta opción analiza si el contenido está escrito con el orden correcto mediante un autómata de pila, como resultado se imprime en consola una tabla de tres columnas donde se indica el contenido de la pila, la entrada y las transiciones.

```

//////////////////////////////// MENU //////////////////////////////////
1. Cargar script
2. Manejo AFD
3. Pila interactiva
4. Diagrama
5. Salir
Ingrese la opcion a seleccionar
3
//////////////////////////////// AP //////////////////////////////////



|   |      |   |   |         |   |   |          |   |
|---|------|---|---|---------|---|---|----------|---|
| . | PILA | . | . | ENTRADA | . | . | TRANCION | . |
|---|------|---|---|---------|---|---|----------|---|



|     |                                                                                                          |
|-----|----------------------------------------------------------------------------------------------------------|
| [ ] | tk_let , tk_Identificador , tk_Igual , tk_true , tk_PuntoYComa , tk_if , tk_ParentesisA , tk_true , tk_P |
|-----|----------------------------------------------------------------------------------------------------------|


```

Si el archivo está escrito de forma correcta se muestra un mensaje indicando que la entrada es válida y se genera un reporte en HTML con una copia de la tabla mostrada en consola.

```

[ '#' ]      2, epsilon, # ; 3 , epsilon

----- ENTRADA VALIDA -----

Desea abrir el reporte ReporteAutomataPila? S/N
S

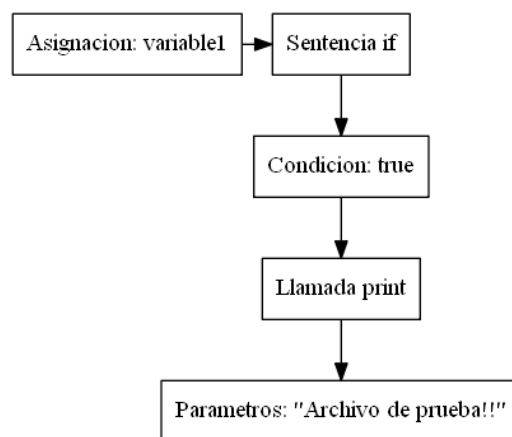
```

ReporteAutomataPila			
No	PILA	ENTRADA	TRANCION
1	[[]]	[tk_let , tk_Identificador , tk_Igual , tk_true , tk_PuntoYComa , tk_if , tk_ParentesisA , tk_true , tk_ParentesisC , tk_LlaveA , tk_Identificador , tk_ParentesisA , tk_Cadena , tk_ParentesisC , tk_PuntoYComa , tk_LlaveC , ']	['0 , epsilon , epsilon ; 1 , #']
2	[['#']]	[tk_let , tk_Identificador , tk_Igual , tk_true , tk_PuntoYComa , tk_if , tk_ParentesisA , tk_true , tk_ParentesisC , tk_LlaveA , tk_Identificador , tk_ParentesisA , tk_Cadena , tk_ParentesisC , tk_PuntoYComa , tk_LlaveC , ']	['1 , epsilon , epsilon ; 2 , S']
3	[['S' , '#' ]]	[tk_let , tk_Identificador , tk_Igual , tk_true , tk_PuntoYComa , tk_if , tk_ParentesisA , tk_true , tk_ParentesisC , tk_LlaveA , tk_Identificador , tk_ParentesisA , tk_Cadena , tk_ParentesisC , tk_PuntoYComa , tk_LlaveC , ']	['2 , epsilon , S ; 2 , [DECLARACION_VARIABLE , S]']

#### Opción 4: Diagrama:

Esta opción crea una imagen que muestra el flujo del código escrito dentro del archivo cargado.

```
//////////////////////////////// MENU //////////////////////////////////
1. Cargar script
2. Manejo AFD
3. Pila interactiva
4. Diagrama
5. Salir
Ingrese la opcion a seleccionar
4
CREANDO DIAGRAMA...
```



#### Opción 5. Salir:

Esta opción se utiliza para finalizar el programa.

```
//////////////////////////////// MENU //////////////////////////////////
1. Cargar script
2. Manejo AFD
3. Pila interactiva
4. Diagrama
5. Salir
Ingrese la opcion a seleccionar
5
Saliendo...

Process finished with exit code 0
```