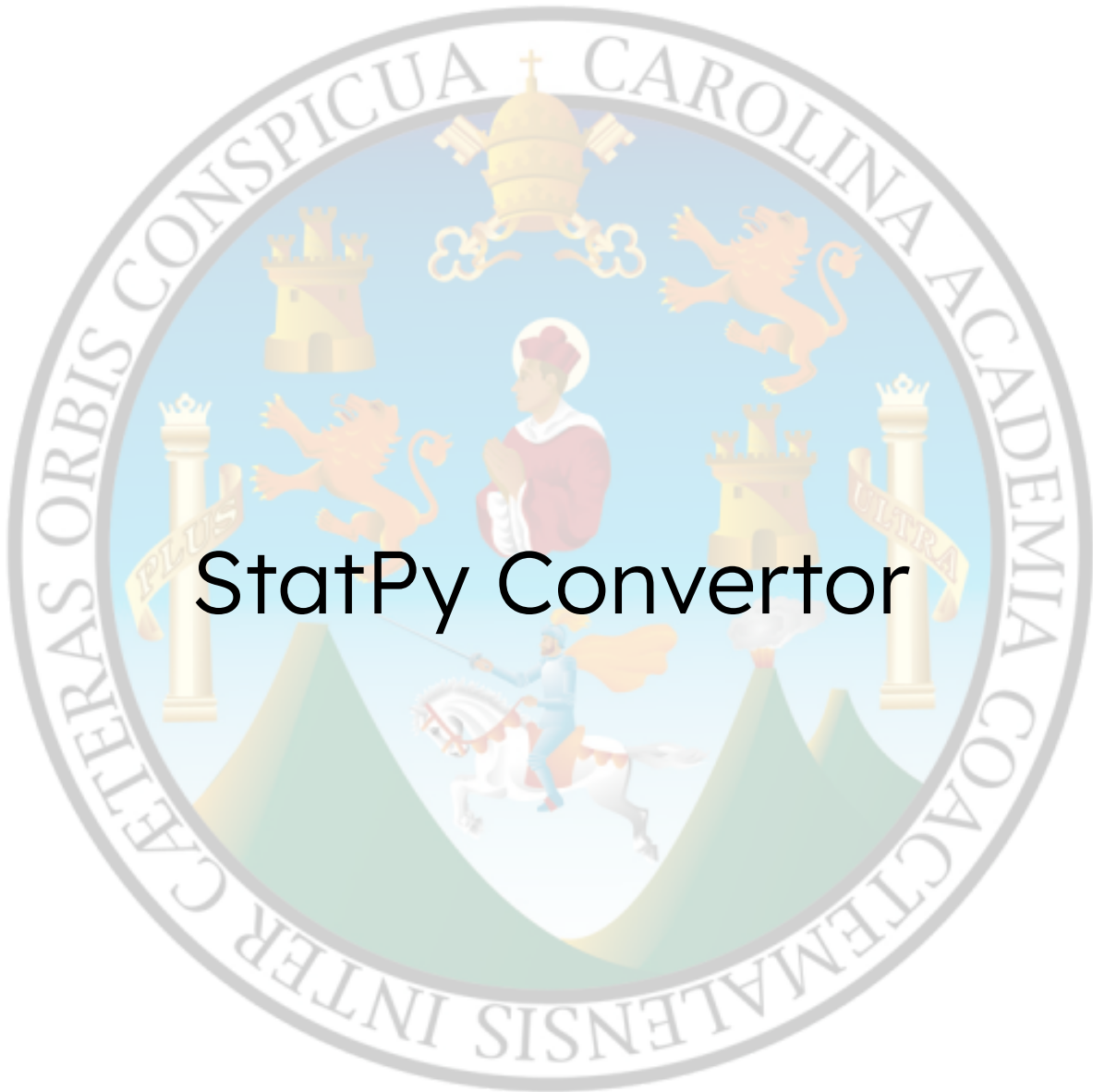


MANUAL DE USUARIO



Contenido

I. Introducción.....	1
1. Objetivo.....	1
2. Requerimiento.....	1
II. Opciones del Sistema.....	2
1. Archivo.....	2
1.1. Abrir.....	2
1.2. Guardar.....	2
1.3. Guardar Como.....	3
2. Analizador.....	3
2.1 StatPy.....	3
2.2 JSON.....	3
3. Ejecutar.....	4
4. Reporte.....	4
4.1 Tokens.....	4
4.2 Errores.....	4
5. Ayuda.....	5
5.1 Manual Técnico.....	5
5.2 Manual de Usuario.....	5
III. Archivo.....	5
1. Estructura de Archivo.....	5

I.Introducción

1. Objetivo

Brindar asistencia al usuario de este programa informático (StatPy Convertor), describiendo las opciones o el funcionamiento del proceso que se muestra en cada una de las pantallas conforme el usuario vaya avanzando en dicho programa.

2. Requerimiento

- Sistema Operativo: Windows 7 o superior
- Procesador mínimo Intel Pentium (800 MHz Intel Pentium)
- Mínimo 1GB en RAM
- IDE Visual Studio Code, o compatible con Python
- Exploradores: Internet Explorer 9 y superior

II.Opciones del Sistema

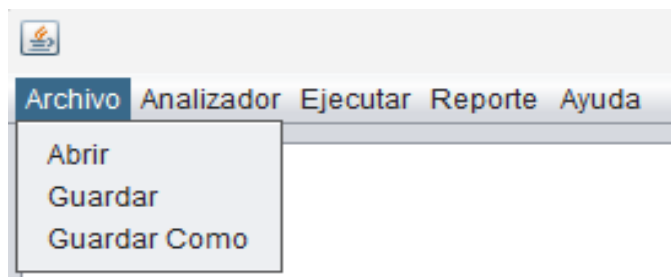
El presente manual está organizado de acuerdo con en el menú de opciones de la siguiente manera:

1. Archivo
 2. Analizador
 3. Ejecutar
 4. Reporte
 5. Ayuda
-

1. Archivo

Al presionar este botón se desplegará una lista con todas las opciones que el usuario puede elegir.

Cada opción es representada por un botón el cual el usuario puede presionar para realizar una acción.



1.1. Abrir

Al hacer clic sobre el botón “Abrir” este mostrará el explorador de archivos para que el usuario seleccione el archivo que desee ejecutar y/o editar.

El programa soporta únicamente archivos con la extensión .sp o .json y pueden ser abiertos varios archivos en la misma ejecución.

Si se intenta abrir un archivo con la extensión incorrecta el programa mostrará un mensaje de advertencia en pantalla y no leerá el archivo seleccionado.

1.2. Guardar

Al presionar el botón “Guardar” el programa guardará el contenido ingresado en el área de texto.

Si existiese un archivo abierto el contenido del área de texto será almacenado en dicho archivo, de no ser así el contenido será almacenado en un archivo nuevo si el usuario así lo desea.

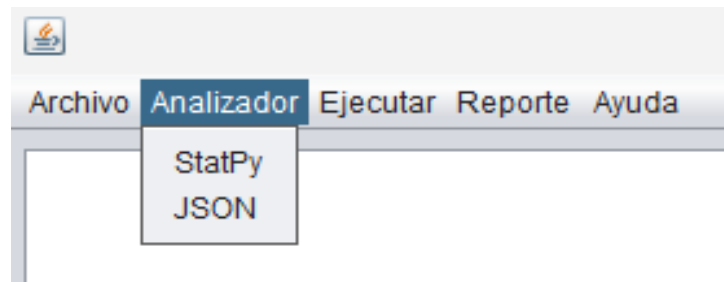
1.3. Guardar Como

Al momento en que el usuario presione el botón “Guarda como” la aplicación mostrará en pantalla el explorador de archivos para que el usuario seleccione la ubicación en la que desea almacenar el archivo e ingrese el nombre que le desea colocar.

2. Analizador

Al presionar este botón se desplegará una lista con todas las opciones que el usuario puede elegir.

Cada opción es representada por un botón el cual el usuario puede presionar para realizar una acción.



2.1 StatPy

Al presionar el botón “StatPy” el programa verificará que en el área de texto se encuentre contenido, de ser así la aplicación procederá a analizar todo el texto y al terminar si no encontrase errores en el código ingresado traducirá este a lenguaje python y también realizará las funciones estadísticas ingresadas.

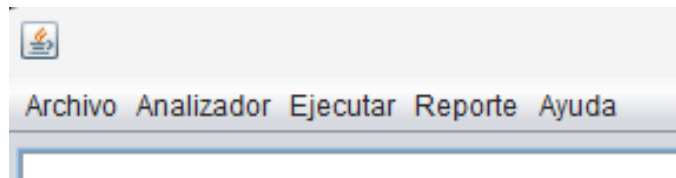
2.2 JSON

Al presionar el botón “JSON” el programa verificará que en el área de texto se encuentre contenido, de ser así la aplicación procederá a analizar todo el texto si existiesen errores el programa mostrará un mensaje informando al usuario que se encontraron errores en el código ingresado. En caso contrario de no encontrarse errores el

programa procederá a almacenar todos los valores en una tabla de símbolos.

3. Ejecutar

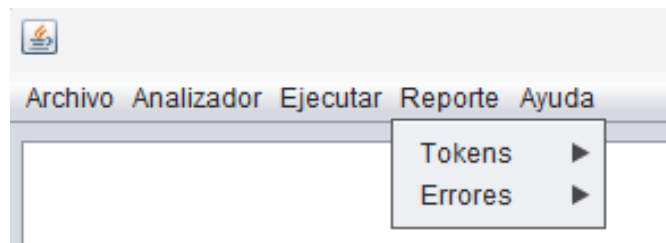
Al hacer clic sobre este botón el programa analizará todo el contenido que encuentre en el área de texto. Debe existir contenido en el área de texto pues de no ser así se mostrará un mensaje de error en la pantalla.



4. Reporte

Al presionar este botón se desplegará una lista con todas las opciones que el usuario puede elegir.

Cada opción es representada por un botón el cual el usuario puede presionar para realizar una acción.



4.1 Tokens

Al hacer clic sobre el botón "Tokens" se redireccionará a una página html en la cual se encuentra una tabla que contiene todos los tokens que el analizador léxico reconoció.

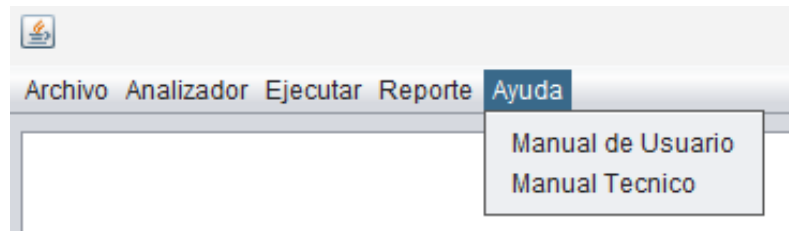
4.2 Errores

Al hacer clic sobre el botón "Errores" se redireccionará a una página html en la cual se encuentra una tabla que contiene todos los errores que el analizador léxico encontró.

5. Ayuda

Al presionar este botón se desplegará una lista con todas las opciones que el usuario puede elegir.

Cada opción es representada por un botón el cual el usuario puede presionar para realizar una acción.



5.1 Manual Técnico

Al presionar este botón el programa muestra el archivo del manual técnico en pantalla.

5.2 Manual de Usuario

Al presionar este botón la aplicación muestra el archivo del manual de usuario en pantalla.

III. Archivo

En esta sección se llevará a cabo cuál es la estructura adecuada que un archivo de entrada debe tener.

1. Estructura de Archivo

Es importante conocer cuál es la estructura correcta que debe tener un archivo de entrada para que el programa no tenga problemas ya sea al traducir el código ingresado a lenguaje python o realizar las gráficas estadísticas.

Si el archivo de entrada no está escrito de forma correcta es posible que el programa lance un error o en caso contrario sea ejecutado pero el resultado puede no ser el que el usuario espera.

Es también de vital importancia que al momento de llamar valores de variables de un archivo .json estas sean llamadas con la estructura correcta.

Ejemplo de archivo .json de entrada:

```
/* ejemplo de carga
de objeto json */

{
    "Titulo": "Grafica Barras",
    "variable1": 4.0,
    "variable2": 5.0,
    "variable3": 3.0
}
```

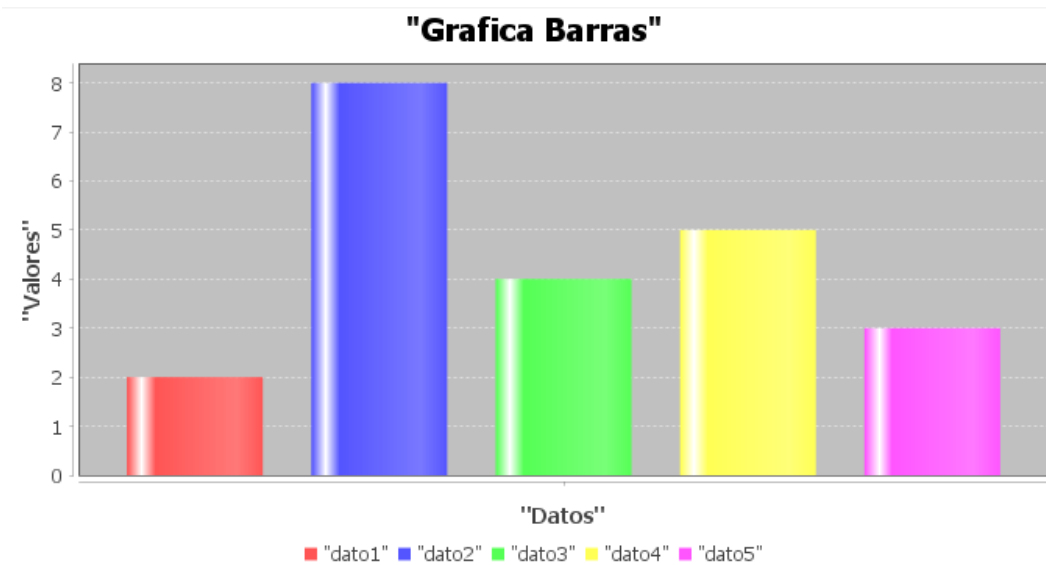
Ejemplo de archivo .sp de entrada:

```
void main ( ){
    int b = 2;
    int a = 1;
    int var1 = 5+8*9;
    if (b > a){
        Console.WriteLine("b mayor que a");
    }else if(a == b){
        Console.WriteLine("a y b son iguales");
    }
    //Funciones para graficar
    void DefinirGlobales(){
        double var1 = 2.0;
        double var2 = 8.0;
        string tituloBarras = ${ NewValor, "graficaBarras.json", "Titulo"};
        string tituloPie = ${ NewValor, "graficaPie.json", "Titulo"};
        string nom1 = "dato1";
        string nom2 = "dato2";
        string nom3 = "dato3";
        string nom4 = "dato4";
        string nom5 = "dato5";
    }
    void GraficaBarras(){
        string Título= tituloBarras;
        string [] Ejex= {nom1, nom2, nom3, nom4, nom5};
        double [] Valores= { var1, var2, ${ NewValor, "graficaBarras.json", "variable1"}, var3, var4, var5 };
        string TítuloX= "Datos";
        string TítuloY= "Valores";
    }
    void GraficaPie(){
        string Título= tituloPie;
        string [] Ejex= {nom1, nom2, nom3, nom4, nom5};
        double [] Valores= { var1, var2, ${ NewValor, "graficaPie.json", "variable1"}, var3, var4, var5 };
    }
    while(a < 10){
        Console.WriteLine("el valor de a es: " + a);
    }
}
```


Resultado de traducción de lenguaje StatPy a Python:

```
def main():  
    b = 2  
  
    a = 1  
  
    var1 = 5 + 8 * 9  
  
    if b > a:  
        print("b mayor que a")  
  
    elif a == b:  
        print("a y b son iguales")  
  
    while a < 10:  
        print("el valor de a es: ", a)  
  
if __name__ == '__main__':  
    main()
```

Resultado de gráficas generadas:



"Grafica Pie"