



PROYECTO #1



MANUAL USUARIO



**EDUARDO JOSUÉ
GONZÁLEZ CIFUENTES**

COMPIADORES 1



201900647

INDICE

Introducción	Pág. 3
Objetivos	Pág. 4
Requerimientos.	Pág. 5
Opciones de la aplicación	Pág. 6
Descripción Lenguaje Statpy y Json	Pág. 8
Recomendaciones	Pág. 12



INTRODUCCIÓN

Bienvenido al Manual de Usuario del Traductor de Lenguaje StatPy a Lenguaje Python con Interfaz Gráfica. Este manual está diseñado para brindarle una experiencia completa y efectiva al utilizar esta poderosa herramienta de traducción de código. Con esta aplicación, podrá convertir con facilidad y precisión código escrito en StatPy a Python, aprovechando las ventajas de ambos lenguajes en sus proyectos de programación y análisis de datos.

En las siguientes secciones, le guiaremos a través de los pasos necesarios para instalar, configurar y utilizar el Traductor. Además, encontrará ejemplos prácticos y casos de uso que le ayudarán a comprender cómo sacar el máximo provecho de esta herramienta. Ya sea que sea un desarrollador experimentado, un estudiante de programación o un entusiasta de la ciencia de datos, este manual está diseñado para brindarle la orientación que necesita para aprovechar al máximo el Traductor de Lenguaje StatPy a Lenguaje Python y simplificar su trabajo en el mundo de la programación y la estadística.



OBJETIVOS

- Guiar al Usuario en la Utilización Efectiva de la Herramienta.
- Fomentar la Comprensión de las Funcionalidades Clave.
- Promover las Mejores Prácticas y la Resolución de Problemas.



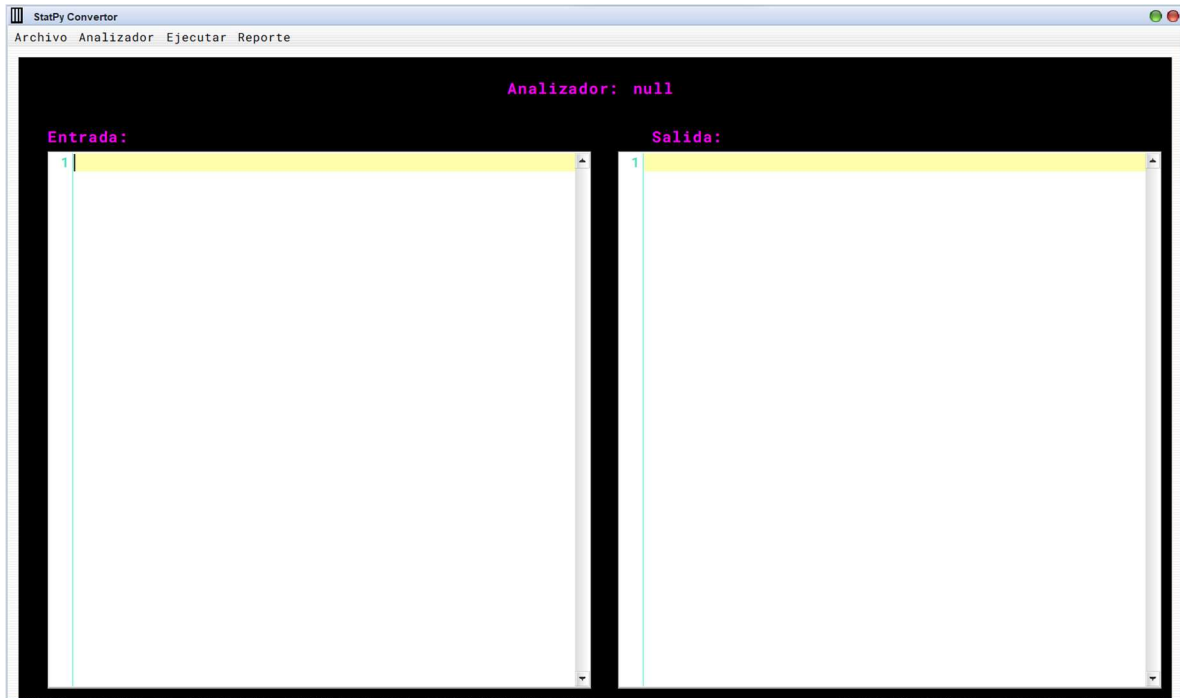
REQUERIMIENTOS

- Computadora portátil o de escritorio.
- Mínimo 4GB de Memoria RAM.
- Windows 10 o superior
- Navegador navegador web.
- JRE 8.1 en adelante
- JDK 8.1 en adelante



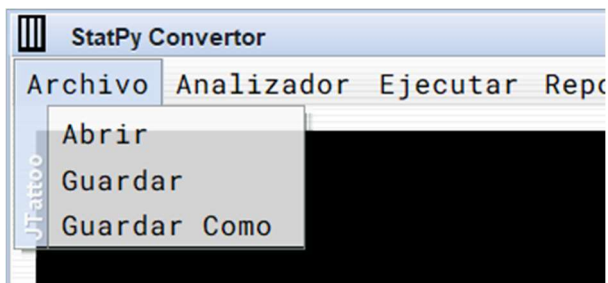
OPCIONES DE LA APLICACIÓN

Acá se puede ver la pantalla principal



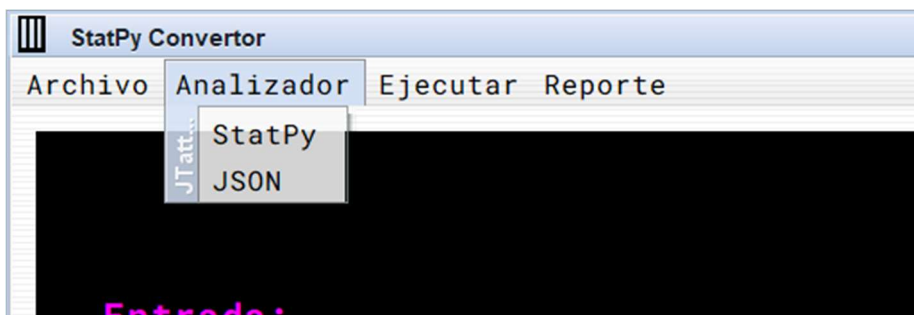
ARCHIVO

Acá tienes las opciones de abrir, guardar y guardar como cualquier tipo de archivo (sp ó json).



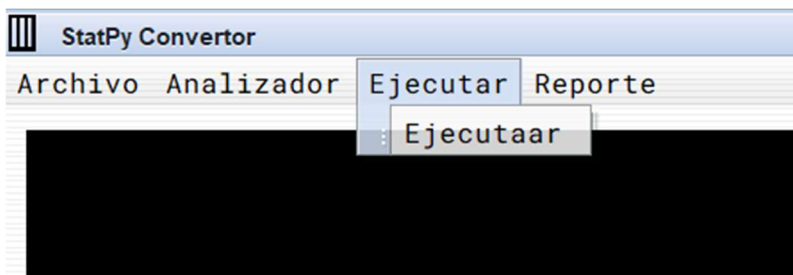
ANALIZADOR

Acá puedes elegir cual de los dos analizadores puedes utilizar.



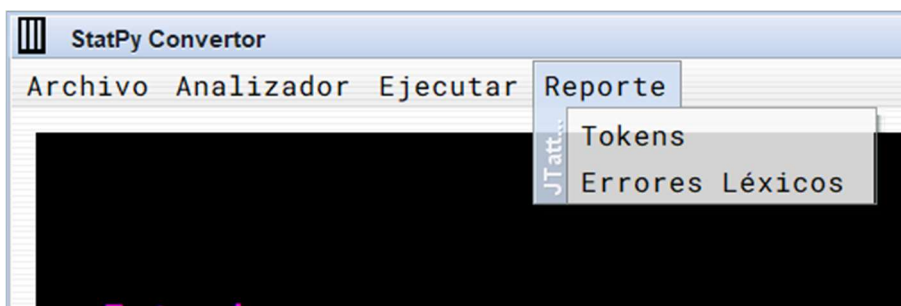
EJECUTAR

Luego de seleccionar tu analizador le puedes dar ejecutar y hará el análisis de json o la traducción a Python.



REPORTE

En esta parte puedes ver los reportes luego de tus análisis ya sea de tokens o errores.



DESCRIPCION LENGUAJE

Acá hay una breve explicación de como es el lenguaje statpy y Python, mas que nada su sintaxis y como deben de venir los archivos.

STATPY

Se debe respetar la sintaxis de statpy para ahorrar errores de compilación.

StatPy	TipoDato ID; TipoDato ID = Expresion;
Salida - Python	id = Expresion

Lenguaje	Ejemplo
StatPy	int var1 = 5+8*9;
Python	var1 = 5+8*9

Lenguaje	Ejemplo
StatPy	void main (){ <Sentencias> }
Python	def main(): <Sentencias> if __name__ == "__main__": main()

Lenguaje	Ejemplo
StatPy	if (b > a){ Console.Write("b mayor que a"); }else if(a == b){ Console.Write("a y b son iguales"); }
Python	if b > a: print("b mayor que a") elif a == b: print("a y b son iguales")



Lenguaje	Ejemplo
StatPy	<pre>switch(valor){ case 1: precio = 55; break; case 2: precio = 25; case 3: precio = 40; default: Console.Write("No válido. Escoja 1, 2, o 3."); }</pre>
Python	<pre>def switch(case, precio): switcher = { 1: precio = 55, 2: precio = 25, 3: precio = 40, 4: print("No válido. Escoja 1, 2, o 3."), }</pre>

Lenguaje	Ejemplo
StatPy	<pre>for (int a=0; a<10; a++){ Console.Write("el valor de a es: " + a); }</pre>
Python	<pre>for a in range(1,10): print("el valor de a es: ", a)</pre>

Lenguaje	Ejemplo
StatPy	<pre>while(a < 10){ Console.Write("el valor de a es: " + a); }</pre>
Python	<pre>while a < 10 : print("el valor de a es:", a)</pre>

Lenguaje	Ejemplo
StatPy	<pre>int a = 1; do { Console.Write("el valor de a es: " + a); } while(a < 5);</pre>
Python	<pre>i = 1 while True: print("el valor de a es: ", a) a = a + 1 if (a < 5): break</pre>

Entrada - StatPy	Console.Write("el valor de a es: " + a);
Salida - Python	print("el valor de a es: ", a)

Operación	Entrada - StatPy	Salida - Python
Suma	+	+
Resta	-	-
Multiplicación	*	*
División	/	/

Operación	Entrada - StatPy	Salida - Python
Mayor	>	>
Menor	<	<
Mayor igual	>=	>=
Menor igual	<=	<=
Igual	==	==

Operación	Entrada - StatPy	Salida - Python
AND	&&	and
OR		or
NOT	!	not

JSON

Se debe respetar la sintaxis de los archivos Json para ahorrar errores de compilación.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The editor contains a JSON object with the following text:

```
/*  
segundo objeto json para archivo medio  
*/  
{  
  "name1": "Luis",  
  "var1": 70.0,  
  "var2": 9.8,  
  "var3": 6.0 |  
}
```

RECOMENDACIONES

- **Realizar Pruebas de Pequeña Escala:** Antes de aplicar la traducción a proyectos de gran envergadura, se recomienda realizar pruebas con fragmentos de código más pequeños. Esto le permitirá familiarizarse con el proceso de traducción y verificar la precisión de los resultados antes de realizar cambios en proyectos completos.
- **Guardar una Copia de Seguridad del Código Fuente Original:** Siempre es aconsejable mantener una copia de seguridad del código fuente original en StatPy antes de realizar la traducción. Esto garantiza que pueda revertir los cambios si es necesario y le proporciona un punto de referencia para comparar con el código traducido.
- **Consultar la Documentación y la Comunidad:** Si encuentra desafíos o preguntas durante el proceso de traducción, no dude en consultar la documentación proporcionada en este manual o buscar ayuda en la comunidad de usuarios. Puede ser beneficioso compartir experiencias y obtener orientación de otros usuarios que hayan trabajado con la herramienta. La colaboración puede ser una fuente valiosa de conocimiento y soluciones.

