

# MANUAL DE USUARIO

## **PLATAFORMA DE SALUD PROYECTO DEL CURSO IPC2: APLICACIÓN WEB CON JAVA Y ANGULAR**

# ¿QUÉ ES ESTE PROYECTO?

¡Bienvenido al Manual de Usuario del proyecto del Analizador Léxico y Sintáctico (limitado) Simulado de Python! Este proyecto ofrece una implementación sencilla de un analizador léxico y sintáctico que emula el comportamiento del lenguaje de programación Python. El proyecto se ha dividido en dos partes, el backend y el frontend, para facilitar su comprensión y uso. A continuación, se detallan los pasos para poner en marcha este analizador léxico:

## 1. Requisitos Previos:

Antes de comenzar, asegúrese de tener instalados los siguientes componentes:

Node.js:

Puede descargarse e instalarlo desde [nodejs.org](https://nodejs.org).

## 2. Iniciar el Backend:

El backend del proyecto se ha desarrollado en Java utilizando librerías como JakartaEE para crear una API que atenderá las solicitudes de los clientes.

## 3. Iniciar el Frontend:

La interfaz de usuario del proyecto se ha creado utilizando el framework JavaScript Vue.js. Para ejecutarla, seguir estos pasos:

- Descargar el proyecto y descomprimirlo en su ubicación preferida.
- Abrir una terminal y navegar hasta la carpeta del proyecto frontend.
- Ejecute el comando `npm install` para instalar las dependencias necesarias.
- Una vez completada la instalación, use `npm run dev` para iniciar el servidor de desarrollo.
- Abrir el navegador web y navegar a la dirección proporcionada por el servidor de desarrollo.

#### 4. Interacción con el Analizador Léxico y Sintáctico:

Una vez que el frontend esté en funcionamiento, tendrá acceso a la interfaz del analizador léxico y sintáctico (limitado) simulado. Ingrese el código en el área designada y este será automáticamente reconocido. El analizador procesa el código y mostrará los resultados (que sean soportados) en la interfaz.

## INTERFAZ Y SUS PARTES

Analizador sintáctico • parser-py • Lenguajes Formales SS 2023

⚙️

Escribe el código aquí:

❌ Limpiar

El cursor está en: L1:C1

```
cadena = "Hola Mundo" # 1
number = 100 # 2
boolean = True # 3

# operadores aritmeticos
suma = 10 + 10 # 4
resta = 10 - 10 # 5
multiplicacion = 10 * 10 # 6
division = 10 / 10 # 7
exponente = 10 ** 10 # 8
modulo = 10 % 10 # 9

# operadores de asignacion
number += 10 # 10
number -= 10 # 11
number *= 10 # 12
number /= 10 # 13
number **= 10 # 14
number %= 10 # 15

metodo1()
metodo1()
metodo1()

metodo2(d()-2)
metodo1(213)
metodo2(123-3123)
```

📄 LEER ARCHIVO

⚙️ ANALIZAR

🔍 Tabla de símbolos de sintaxis

Symbol	Tipo	Valor	Línea	Columna
<code>cadena</code>	VARIABLE	"Hola Mundo"	1	7
<code>number</code>	VARIABLE	100	2	7
<code>boolean</code>	VARIABLE	True	3	8
<code>suma</code>	VARIABLE	10+10	6	5
<code>resta</code>	VARIABLE	10-10	7	6

Items per page: 

5

 1-5 of 22 < > >|

🔍 Tabla de llamadas a métodos

Total de métodos encontrados: 3

🕒 Tabla de errores de sintaxis


🕒 Errores / Tokens inválidos:

## → EDITOR DE TEXTO

La aplicación proporciona un pequeño y simple editor de texto con resaltado de sintaxis a medida de que se va escribiendo. En la parte superior muestra la información del cursor y su posición. También se proporciona un pequeño botón para limpiar el editor y comenzar de nuevo.

Escribe el código aquí:

 Limpiar

El cursor está en: L1:C1 

```
cadena = "Hola Mundo" # 1
number = 100 # 2
boolean = True # 3

# operadores aritmeticos
suma = 10 + 10 # 4
resta = 10 - 10 # 5
multiplicacion = 10 * 10 # 6
division = 10 / 10 # 7
exponente = 10 ** 10 # 8
modulo = 10 % 10 # 9

# operadores de asignacion
number += 10 # 10
number -= 10 # 11
number *= 10 # 12
number /= 10 # 13
number **= 10 # 14
number %= 10 # 15

metodo1()
metodo1()
metodo1()

metodo2(d()-2)
metodo1(213)
metodo2(123-3123)
```

## → TABLAS DE INFORMACIÓN DE ANÁLISIS SINTÁCTICO

Este componente muestra información en formato de tabla acerca de la información que se ha obtenido a través del (limitado) análisis sintáctico

La primera tabla mostrará información acerca de los símbolos que se han encontrado durante el análisis. El estado de este analizador sólo permite detectar asignaciones de variables y llamadas a métodos válidos.

La segunda tabla mostrará información sobre las llamadas a métodos y su posición. También un contador de los métodos que se han detectado.

La tercera tabla muestra información acerca de cualquier error sintáctico que se haya detectado.

La última tabla muestra información sobre los tokens inválidos que se hayan encontrado durante la primera fase de análisis léxico.

- **Tabla de símbolos de sintaxis**

🔗 Tabla de simbolos de sintaxis ^				
Symbol	Tipo	Valor	Linea	Columna
cadena	VARIABLE	"Hola Mundo"	1	7
number	VARIABLE	100	2	7
boolean	VARIABLE	True	3	8
suma	VARIABLE	10+10	6	5
resta	VARIABLE	10-10	7	6
Items per page: 5 ▾ 1-5 of 22  < < > >				
🔗 Tabla de llamadas a metodos Total de métodos encontrados: 3 ▾				
🔗 Tabla de errores de sintaxis ▾				
🔗 Errores / Tokens invalidos: ▾				

- **Tabla de llamadas a métodos**

## Tabla de símbolos de sintaxis

## 📄 Tabla de llamadas a metodos

Total de métodos encontrados: **3**

Metodo	Veces llamado	Linea ultima llamada
metodo1	4	26
d	1	25
metodo2	2	27

Items per page:

5 ▼

1-3 of 3

⏪

<

>

⏩

## ❗ Tabla de errores de sintaxis

 **Errores / Tokens invalidos:**

- **Tabla de errores de sintaxis**

⚙️ <b>Tabla de símbolos de sintaxis</b> <span>▼</span>		
⚙️ <b>Tabla de llamadas a métodos</b> <span>Total de métodos encontrados: 3</span> <span>▼</span>		
⚠️ <b>Tabla de errores de sintaxis</b> <span>^</span>		
Error	Linea	Columna
Se esperaba un identificador	29	4
Items per page: <span>5 ▼</span> 1-1 of 1 <span>⏪</span> <span>⏩</span> <span>⏴</span> <span>⏵</span>		
⚠️ <b>Errores / Tokens invalidos:</b> <span>▼</span>		

- **Tabla de tokens invalidos**

⌵ <b>Tabla de simbolos de sintaxis</b> ⌵			
⌵ <b>Tabla de llamadas a metodos</b> Total de métodos encontrados: 3 ⌵			
ⓘ <b>Tabla de errores de sintaxis</b> ⌵			
ⓘ <b>Errores / Tokens invalidos:</b> ^			
Lexema	Tipo	Linea	Columna
\$id	INVALID_UNIDENTIFIED	29	4
Items per page: 5 ▾ 1-1 of 1  < < > >			



## → ACCIONES DE CÓDIGO

Se proporciona dos acciones que pueden ser realizadas sobre el editor de texto, que son, la carga de archivos para ingresar código por medio de un archivo y su contenido, y el análisis manual del mismo en caso haya un problema



LEER ARCHIVO



ANALIZAR

## → CARGA DE ARCHIVOS

Al seleccionar la opción de *Leer Archivo*, se mostrará un diálogo pidiendo que se seleccione desde la máquina local un archivo en formato de texto plano o con extensión .py para poder importar su contenido al editor de texto y analizarlo



## Leer desde archivo



Aquí puedes seleccionar un archivo de texto plano para leerlo y analizarlo



Archivo de texto

d1.png

 **Tipo de archivo detectado:**

 image/png

 **Ocurrió un error al leer el archivo:**

El tipo de archivo parece no ser valido, solo se permiten archivos de texto plano.

USAR ARCHIVO