MANUAL USUARIO PROYECTO 2

JavaBridge: Traductor de Lenguajes Java a Python

Mario Rodrigo Balam

Correo electrónico: 2908263140708

Carné: 202200147

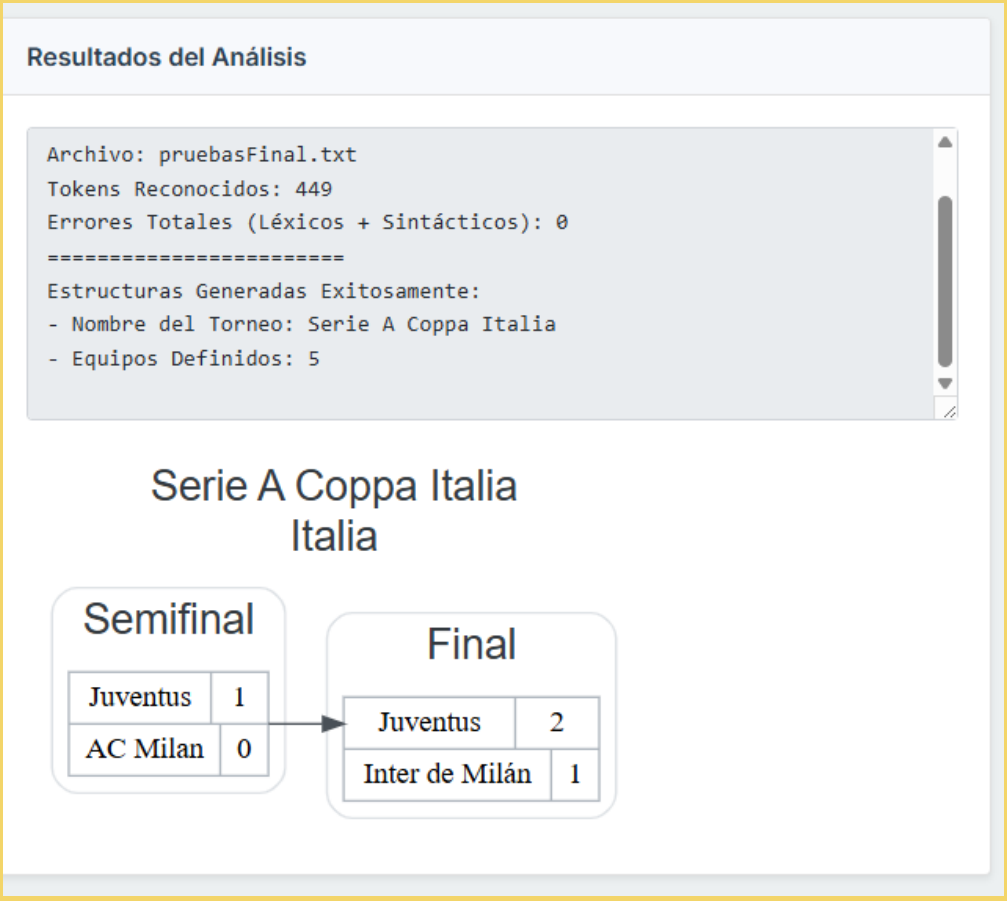
Curso: Lenguajes Formales

Sección: A-

Ing Vivian Damaris Campos González

Aux: Carlos Javier Cox Bautista

TABLA DE CONTENIDO



INTRODUCCION

¡JavaBridge es un proyecto académico desarrollado para el curso de Lenguajes Formales y de Programación** de la Facultad de Ingeniería, Universidad de San Carlos de Guatemala. El objetivo principal de este proyecto es construir un traductor que convierta un subconjunto definido del lenguaje de programación Java a su equivalente en Python.

La herramienta está diseñada como una aplicación web interactiva que permite al usuario escribir o cargar código Java, analizarlo, y ver la traducción resultante en tiempo real. Una de las restricciones fundamentales del proyecto es que tanto el **analizador léxico** como el **analizador sintáctico** deben ser implementados de forma manual, sin el uso de librerías o generadores automáticos (como ANTLR) ni expresiones regulares para el procesamiento de tokens.

OBJETIVO GENERAL

El objetivo principal es proporcionar una plataforma que:

1. Realice el análisis léxico de archivos que contengan definiciones de torneos deportivos
2. Identifique y clasifique tokens según una gramática definida
3. Detecte y reporte errores léxicos con precisión
4. Genere estructuras de datos organizadas a partir de la información procesada
5. Ofrezca reportes detallados y visualizaciones intuitivas de los datos analizados

TECNOLOGIAS UTILIZADAS

IV. Especificación técnicas

- **Frontend**: HTML5, CSS3, JavaScript (ES6+)
- **Framework CSS**: Bootstrap 5.0.2,
- **Backend**: JavaScript
- **Almacenamiento**: Manejo de archivos locales mediante File API

V. Estructura del proyecto

```
javabridgetranslator/
├── docs/
│   └── (Aquí van mi manual técnico, diagramas, etc.)
│
├── public/
│   ├── css/
│   │   └── style.css
│   ├── js/
│   │   └── app.js          // Lógica de la interfaz (botones, DOM)
│   └── index.html          // La interfaz gráfica
│
├── src/
│   ├── analyzer/
│   │   ├── Lexer.js        // Clase para el Analizador Léxico
│   │   ├── Parser.js       // Clase para el Analizador Sintáctico
│   │   └── Translator.js   // Clase para el Traductor
│   │
│   └── utils/
│       └── ReportGenerator.js // Para crear el HTML de los reportes
│
├── test-cases/
│   ├── caso1_valido.java
│   └── caso2_error_lexico.java
│
├── .gitignore
├── package.json
└── server.js               // Nuestro servidor web con Node.js y Express
```

REQUISITOS DEL SISTEMA

- Lenguaje de Programación: JavaScript (ECMAScript 6+).
- Entorno de Ejecución: Navegador web (para la interfaz) y Node.js
- Interfaz Gráfica: HTML5, CSS3.
- Control de Versiones: Git y GitHub.
- Herramientas de Diagramación: PlantUML / Mermaid.

Guía de Uso Paso a Paso

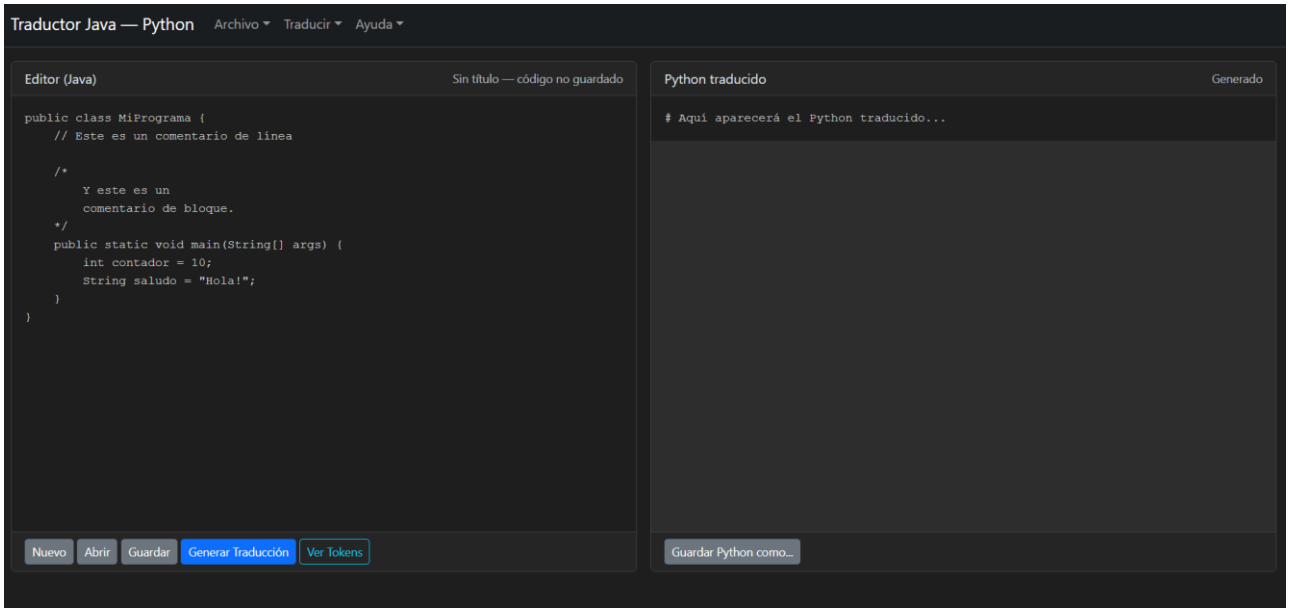
Paso 1: Cargar un Archivo de Java

La primera acción es cargar el archivo .java que contiene el código de java para poder traducirlo..

1. En la barra de navegación inverior, localiza el selector de archivos.
2. Haz clic en **"Seleccionar archivo"** (o similar). Se abrirá el explorador de archivos de tu sistema.
3. Navega hasta tu archivo .txt, selecciónalo y haz clic en "Abrir".
4. Finalmente, haz clic en el botón azul **"Cargar Archivo"**.

Al cargar el archivo, verás que la información de este (nombre y tamaño) aparece en la tarjeta "Definición del Torneo", y su contenido se mostrará en el área de texto.

.



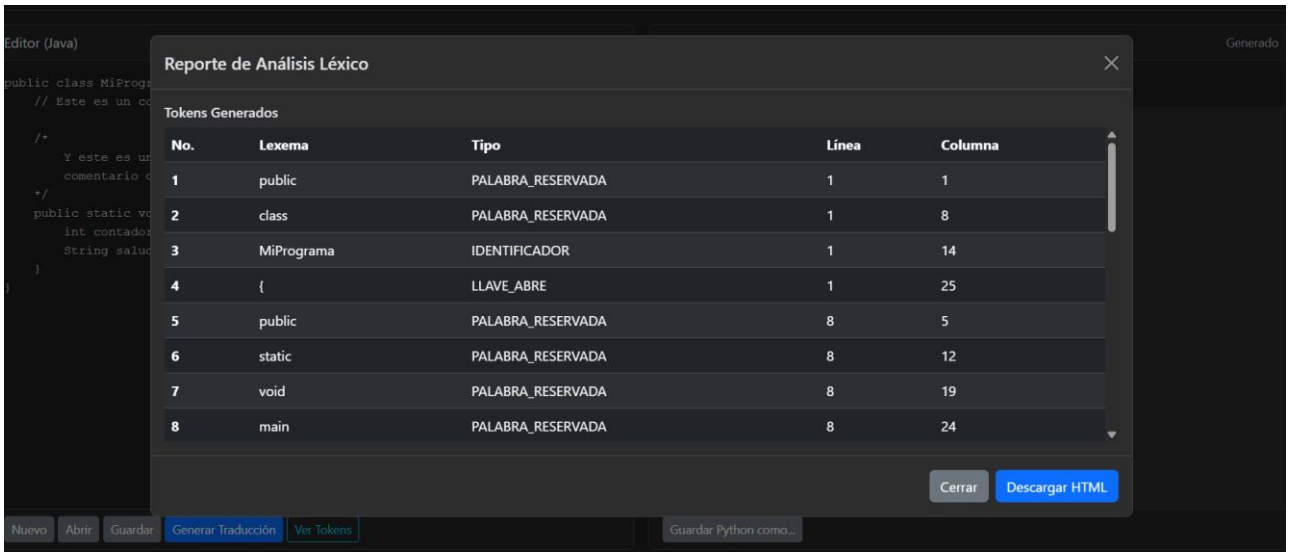
Paso 2: Ver Tokens.

Una vez cargado el archivo, el siguiente paso es ver los tokens que analice.

1. El botón **"Analizar tokens"** en la barra de navegación superior se habrá habilitado.

2. Haz clic en este botón.

La aplicación procesará el archivo en segundos. Si el análisis es exitoso y no encuentra errores, el área de "Resultados del Análisis" aparecerá con un resumen de las estructuras generadas. Si se encuentran errores, se mostrará un alert y el resumen indicará que el análisis finalizó con problemas.

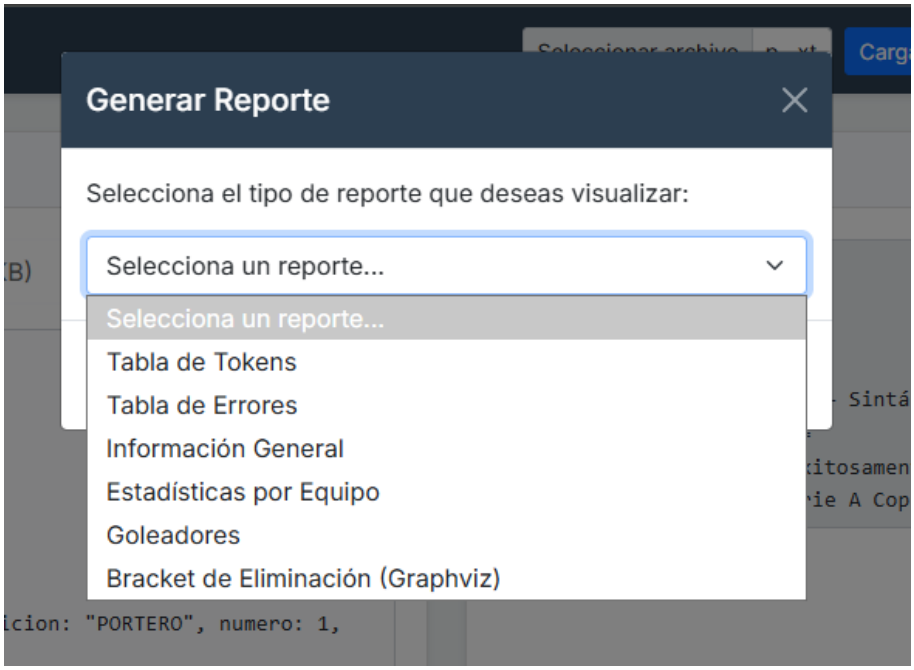


Paso 3: Generar Reportes

Después de un análisis, puedes visualizar los datos en detalle a través de varios reportes.

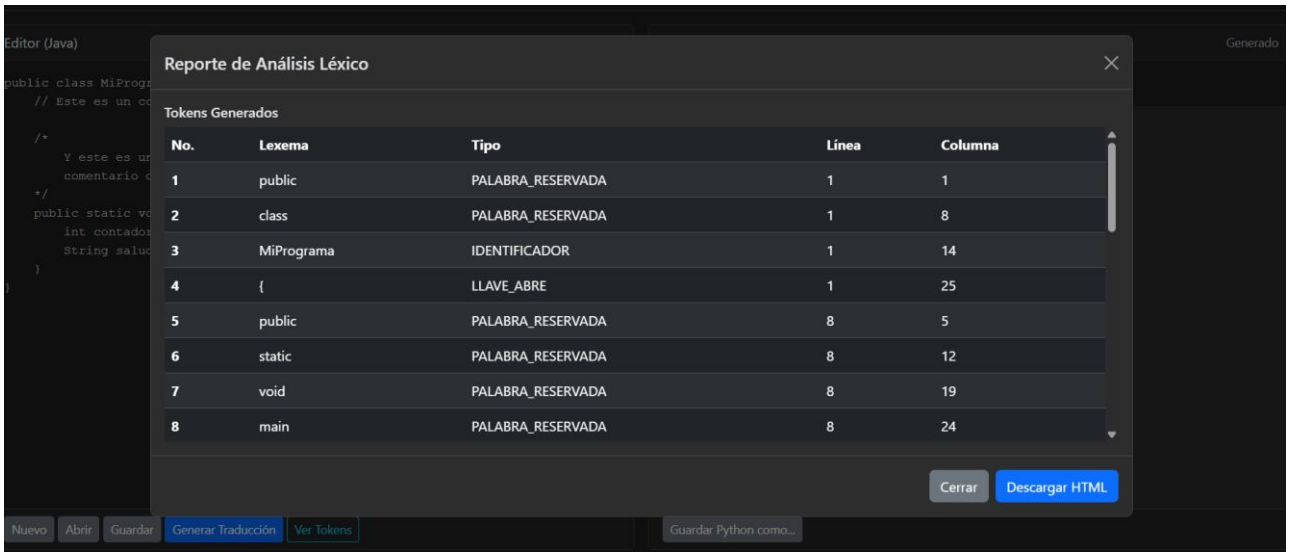
1. Haz clic en el botón **"Generar Reporte"** en la barra de navegación. Aparecerá una ventana emergente (modal).
2. En el menú desplegable, **selecciona el reporte** que deseas ver (ej. "Tabla de Tokens", "Estadísticas por Equipo", etc.).
3. Haz clic en el botón azul **"Generar"**.

La ventana se cerrará y el resultado del reporte aparecerá en la tarjeta "Resultados del Análisis". Puedes generar diferentes reportes cuantas veces quieras sin necesidad de volver a analizar el archivo.



Ejemplo 1: Reporte de Tabla de Tokens

Este reporte es útil para desarrolladores, ya que muestra cómo la aplicación dividió el archivo en sus componentes léxicos básicos.



CONCLUSIONES

1. **La Separación de Fases es Clave para la Robustez:** El desarrollo de TourneyJS demostró la importancia crítica de separar el análisis léxico del sintáctico. Al tener un scanner enfocado únicamente en la tokenización y un parser enfocado en la gramática, fue posible depurar y resolver problemas complejos (como los bucles infinitos) de manera aislada y eficiente, lo que habría sido casi imposible en un sistema monolítico.
2. **La Importancia de un Léxico Bien Definido:** Se concluye que la precisión y consistencia en la definición de los tipos de token en el analizador léxico son fundamentales para la simplicidad del analizador sintáctico. La decisión de clasificar las palabras clave en "estructurales" y de "propiedad" directamente en el scanner simplificó enormemente la lógica del parser, haciéndolo más legible y fácil de mantener.
3. **El Descenso Recursivo como Herramienta Didáctica y Efectiva:** La implementación de un parser de descenso recursivo manual, aunque más laboriosa que usar un generador automático, proporcionó una comprensión profunda de cómo se aplican las reglas gramaticales para construir un árbol de sintaxis abstracto. Este enfoque resultó ser perfectamente adecuado y eficiente para la complejidad del lenguaje de torneos definido.