



# Manual de Usuario

## JavaBridge: Traductor Java a Python

Angel Raúl Herrera Chilel  
Carnet: 202402587  
Sección: B  
25 de Octubre de 2025

# Índice

## Índice

|   |          |
|---|----------|
| <b>Índice</b> . . . . .                         | <b>1</b> |
| <b>1. Introducción</b> . . . . .                | <b>2</b> |
| <b>2. Repositorio</b> . . . . .                 | <b>2</b> |
| <b>3. Requerimientos del Sistema</b> . . . . .  | <b>2</b> |
| 3.1. Entorno de Ejecución . . . . .             | 2        |
| 3.2. Configuración Inicial . . . . .            | 2        |
| <b>4. Funcionamiento del Programa</b> . . . . . | <b>3</b> |
| 4.1. Interfaz Principal . . . . .               | 3        |
| 4.2. Menú Archivo . . . . .                     | 3        |
| 4.3. Menú Traducir . . . . .                    | 3        |
| 4.4. Menú Ayuda . . . . .                       | 4        |
| <b>5. Conclusiones</b> . . . . .                | <b>4</b> |
| <b>6. Recomendaciones</b> . . . . .             | <b>4</b> |
| <b>7. Imágenes</b> . . . . .                    | <b>4</b> |

# 1 Introducción

El presente manual de usuario describe el funcionamiento del sistema *JavaBridge*, una aplicación web diseñada para traducir código de un subconjunto de Java a Python. Este programa permite a los usuarios escribir o cargar código Java, analizarlo léxica y sintácticamente, y generar el código Python equivalente si la sintaxis de entrada es correcta. El propósito de este manual es guiar al usuario en la instalación, configuración y uso del sistema, asegurando una experiencia eficiente y sin complicaciones. Este documento está dirigido a usuarios (estudiantes, programadores) que deseen utilizar la herramienta de traducción.

## 2 Repositorio

Repositorio en GitHub: [Enlace al código fuente](#)

## 3 Requerimientos del Sistema

Para garantizar el correcto funcionamiento de *JavaBridge*, se deben cumplir los siguientes requerimientos:

### 3.1 Entorno de Ejecución

- **Navegador Web:** Google Chrome, Mozilla Firefox, Microsoft Edge o Safari (versiones recientes).
- **Entorno de Servidor (Backend):** Node.js v18 o superior.
- **Intérprete de Python:** Python v3.x (para la función de simulación).

### 3.2 Configuración Inicial

El sistema consta de dos partes: un **backend** (servidor) y un **frontend** (interfaz web). Ambos deben estar en ejecución simultáneamente.

1. Abra una terminal en la carpeta **backend** del proyecto.
2. Ejecute `npm install` para instalar las dependencias (Express, CORS).
3. Ejecute `npm run dev` para iniciar el servidor (usualmente en `http://localhost:4000`).
4. Abra una **segunda terminal** en la carpeta **frontend** del proyecto.
5. Ejecute `npm install` para instalar las dependencias (Vite).
6. Ejecute `npm run dev` para iniciar la interfaz web (usualmente en `http://localhost:5173`).
7. Abra la dirección del **frontend** (ej. `http://localhost:5173`) en su navegador web.

## 4 Funcionamiento del Programa

A continuación, se detalla el uso de la interfaz gráfica del sistema.

### 4.1 Interfaz Principal

La aplicación se divide en tres áreas principales:

- **Editor Java (Izquierda):** Área para escribir o pegar el código Java de entrada. Cuenta con numeración de línea y indentación (2 espacios) al presionar la tecla `Tab`.
- **Python Traducido (Derecha):** Área de solo lectura donde aparecerá el código Python resultante tras una traducción exitosa.
- **Consola de Simulación (Abajo):** Muestra los mensajes del sistema. Aquí aparecerán los errores léxicos/sintácticos o la salida de la simulación de Python.

### 4.2 Menú Archivo

- **Nuevo:** Limpia todos los editores y la consola.
- **Abrir (.java):** Abre un explorador de archivos para cargar un archivo `.java` en el `.Editor Java`.
- **Guardar Java:** Descarga el contenido actual del `.Editor Java` como un archivo `.java`.
- **Guardar Python (.py):** Descarga el contenido actual del "Python Traducido" como un archivo `.py`.

### 4.3 Menú Traducir

- **Generar Traducción:** Es la acción principal. Envía el código del `.Editor Java` al servidor para ser analizado.

- **IMPORTANTE:** El código Java debe seguir la estructura obligatoria:

```
public class NombreClase {  
    public static void main(String[] args) {  
        // ... sentencias aqui ...  
    }  
}
```

- **Si es exitoso:** El código Python aparece en el panel derecho y la consola muestra "Traducción generada exitosamente."
  - **Si falla:** El panel de Python queda vacío y la consola muestra la lista de errores léxicos y/o sintácticos encontrados.
- **Ver Reportes:** Abre una nueva pestaña del navegador con los reportes HTML detallados del último análisis, incluyendo la tabla de Tokens, Errores Léxicos y Errores Sintácticos.

- **Simular Ejecución:** Si la traducción fue exitosa, esta acción ejecuta el código Python generado en el servidor y muestra la salida (ej. de la función `print()`) o cualquier error de ejecución de Python en la Consola de Simulación".

## 4.4 Menú Ayuda

- **Acerca de:** Muestra la información del desarrollador y del proyecto.

## 5 Conclusiones

*JavaBridge* ofrece una solución web portátil y eficiente para la traducción de un subconjunto de Java a Python, cumpliendo con los requisitos de análisis léxico y sintáctico manual. La interfaz facilita la carga de archivos, la visualización de resultados y la depuración de errores mediante reportes y simulación.

## 6 Recomendaciones

- **Verifique la estructura:** La causa más común de error sintáctico es no incluir la estructura obligatoria `public class ...` y `public static void main ....`
- **Revise ambas terminales:** Asegúrese de que tanto el servidor `backend` como el `frontend` estén en ejecución y no muestren errores.
- **Use los reportes:** Si la traducción falla, el reporte HTML es la mejor herramienta para ver exactamente qué token o línea causó el problema.

## 7 Imágenes

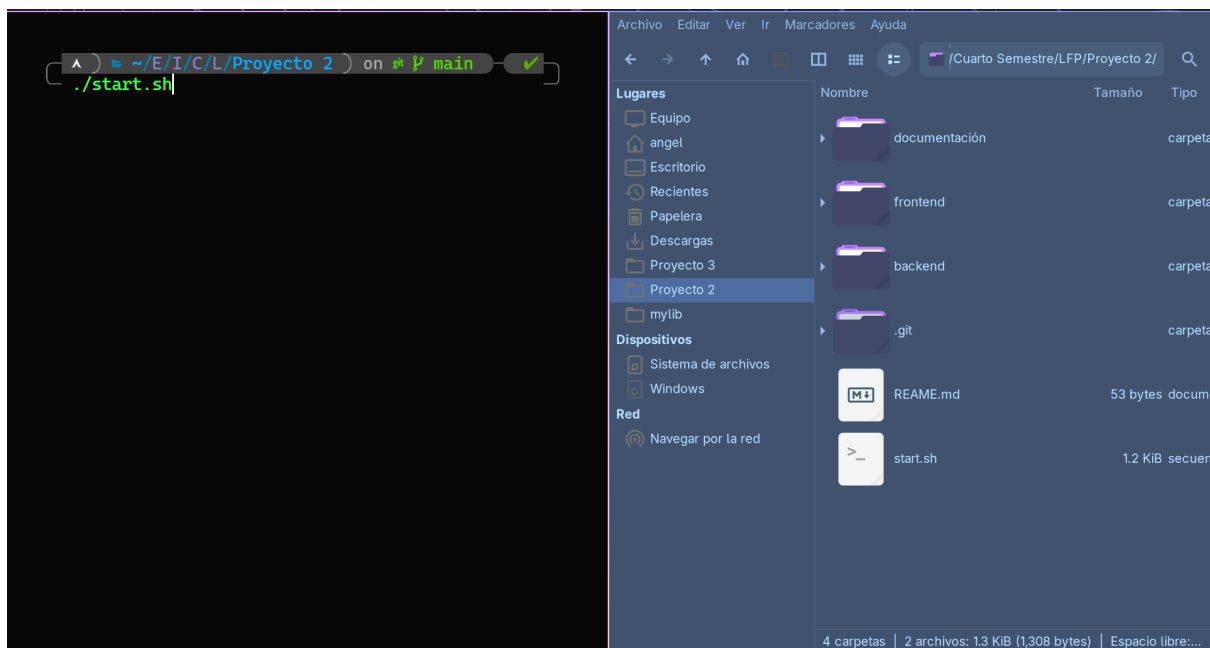


Figura 1: Iniciar proyecto

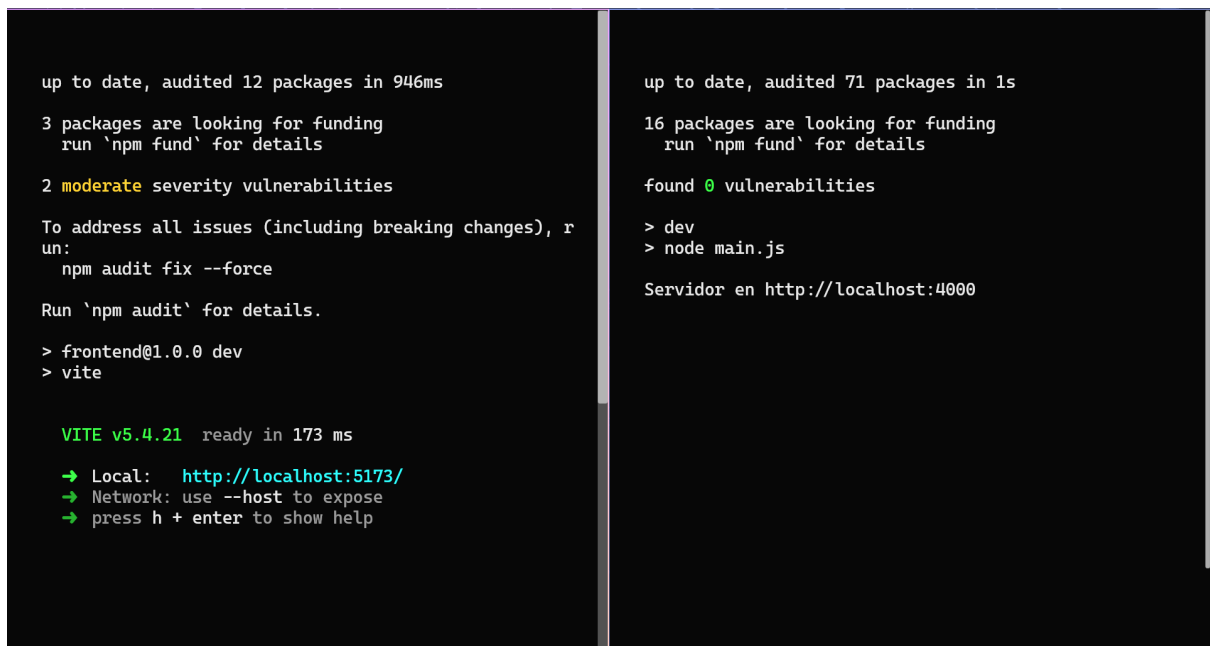


Figura 2: Localhost corriendo en ambos puertos necesarios

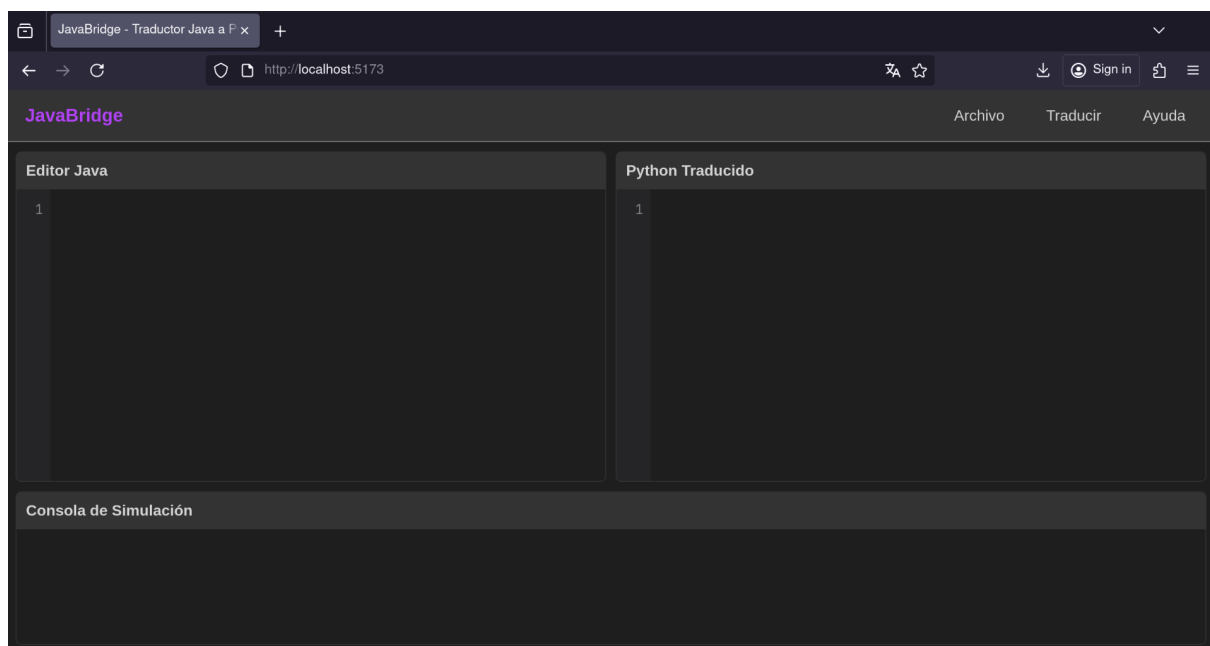


Figura 3: Programa abierto

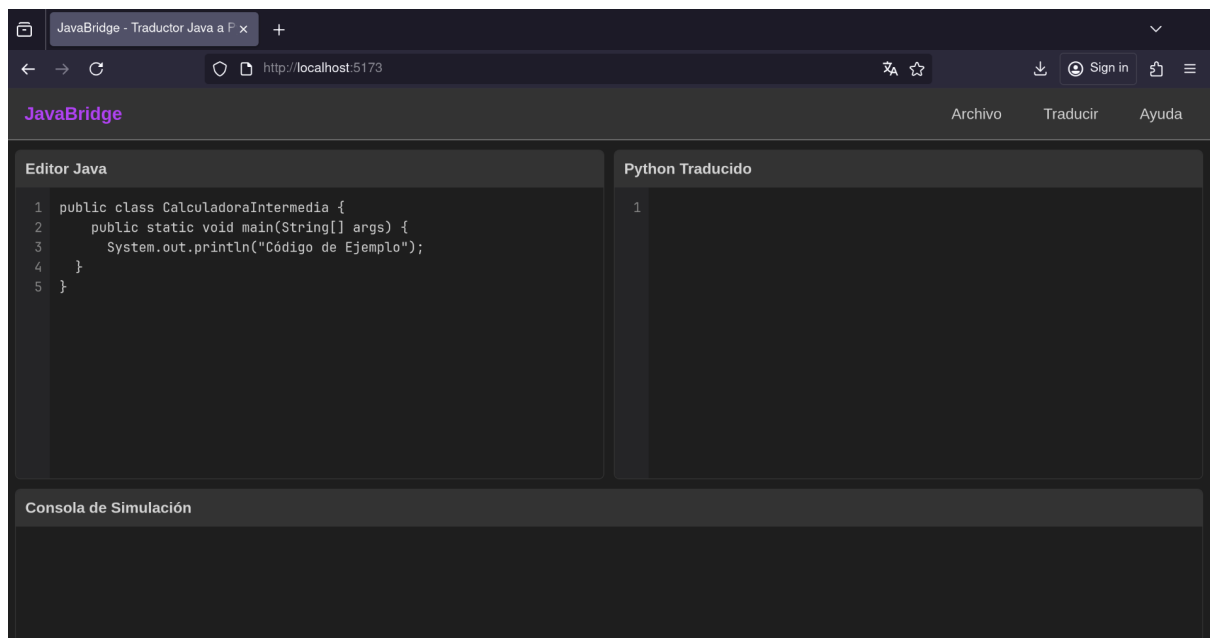


Figura 4: Código Java de Ejemplo

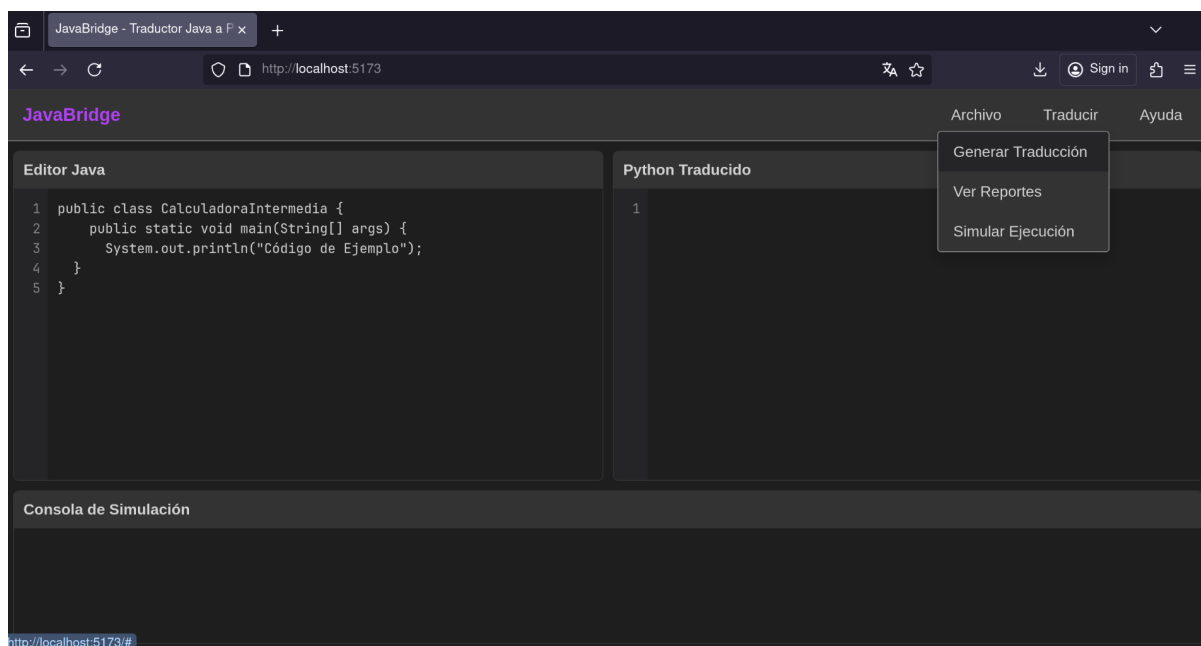


Figura 5: Muestra de botones disponibles

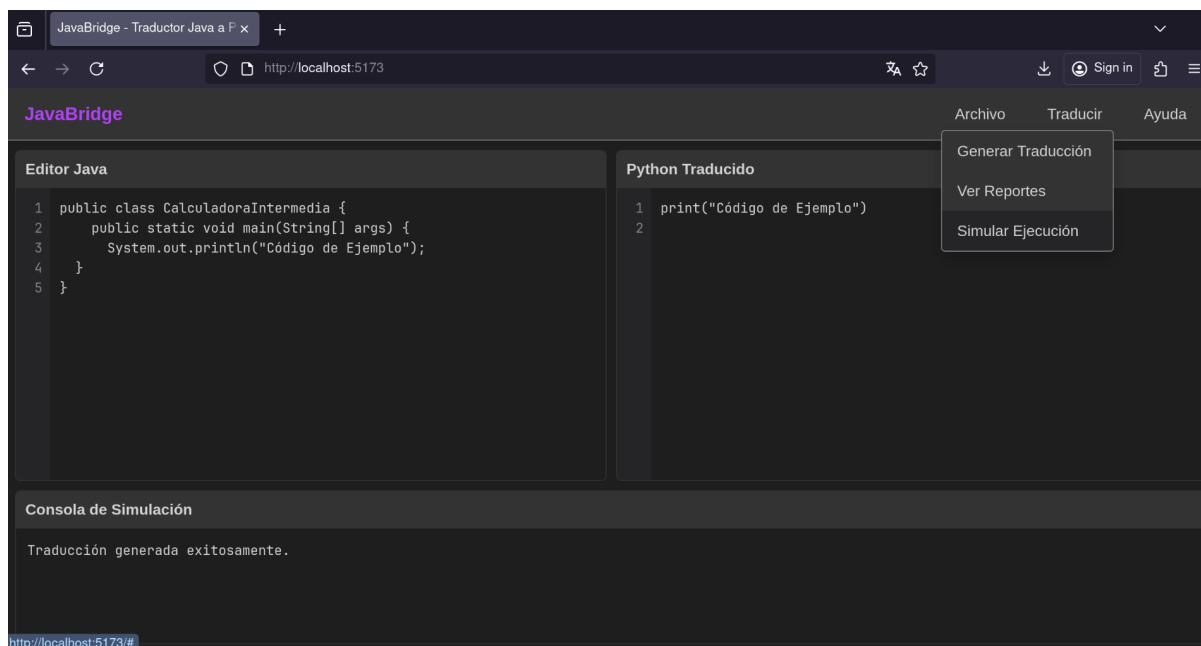


Figura 6: Código Python generado





Figura 7: Vista de los reportes y análisis

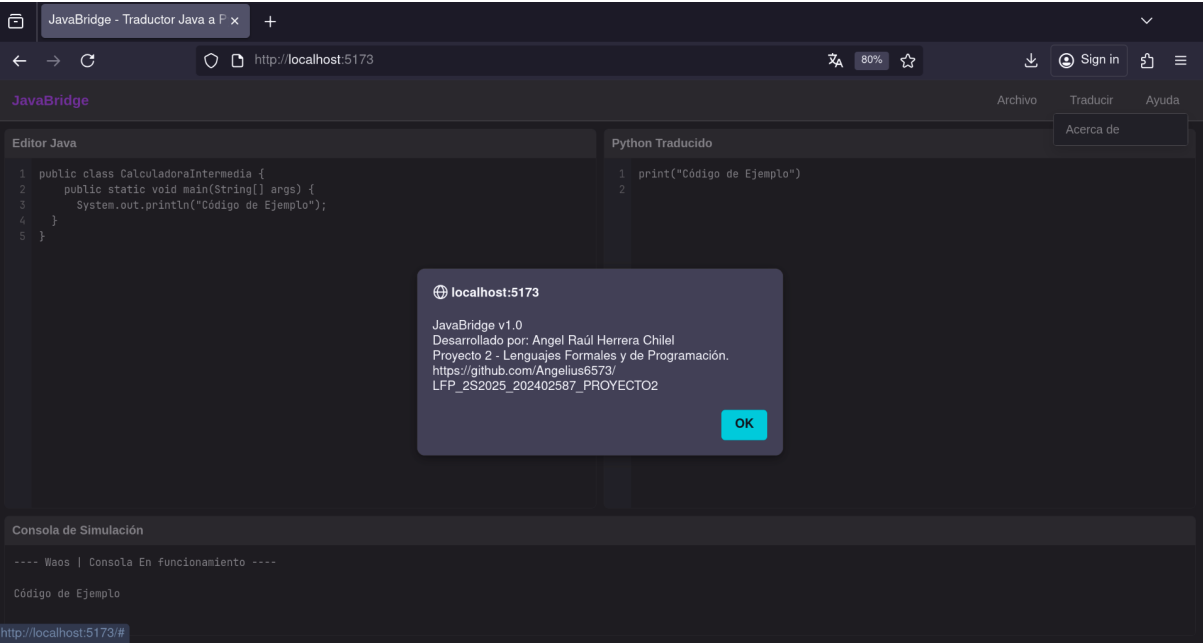


Figura 8: Información del estudiante