Manual de Usuario - Traductor Java → Python

¿Qué es este proyecto?

Un **traductor de código** que convierte programas básicos de Java a Python de forma automática, con interfaz web visual.

Cómo Iniciar el Proyecto

Opción 1: VS Code (Más Fácil)

1. Presiona `Ctrl + Shift + B`
2. Selecciona "**Iniciar Servidor Web**"
3. Abre tu navegador en: `http://localhost:3000`

Opción 2: Terminal

```
npm run server
```

Opción 3: Modo CLI (Sin interfaz)

```
npm start src/ejemplos/test.java
```

Cómo Usar la Interfaz Web

Paso 1: Abrir la Aplicación

Visita `http://localhost:3000` en tu navegador.

Paso 2: Escribir o Cargar Código

Opción A: Escribe tu código Java directamente en el editor izquierdo.

Opción B: Haz clic en **"Cargar ejemplo"** para usar archivos de prueba.

Paso 3: Analizar el Código

Haz clic en el botón **"Analizar Código"**.

Paso 4: Ver Resultados

La aplicación mostrará **4 pestañas**:

Tokens

- Lista de todas las palabras reconocidas del código
- Muestra: tipo, valor, línea y columna
- **Ejemplo:** `PUBLIC`, `class`, `Main`, `{`, etc.

🌳 AST (Árbol Sintáctico)

- Estructura jerárquica del programa
- Visualiza cómo se organiza el código
- **Ejemplo:**

```
Programa
└── Clase
    ├── Nombre: Main
    └── Metodo
        └── Nombre: main
```

❌ Errores

- Muestra errores léxicos o sintácticos
- Indica línea y columna del problema
- **Ejemplo:** `"Se esperaba ';' en línea 5"`

🐍 Código Python

- El código traducido automáticamente
- Listo para copiar y ejecutar
- **Ejemplo:**

```python
def main():
    print("Hola Mundo")

if __name__ == "__main__":
    main()
```

📊 Diagrama de Flujo del Proceso

```
flowchart TD
    A[Usuario escribe código Java] --> B{¿Código válido?}
    B -->|No| C[Mostrar Errores]
    B -->|Sí| D[Análisis Léxico]
    D --> E[Generar Tokens]
    E --> F[Análisis Sintáctico]
    F --> G[Construir AST]
    G --> H[Validación]
    H --> I{¿Estructura correcta?}
```

```
    I -->|No| C
    I -->|Sí| J[Traducción]
    J --> K[Código Python]
    K --> L[Mostrar resultado]
    C --> L
```

🔍 Ejemplo Completo

Código Java de Entrada:

```java
public class Main {
    public static void main(String[] args) {
        int x = 10;
        System.out.println(x);
    }
}
```

Proceso:

1. **Lexer:** Identifica 20 tokens (public, class, Main, {, int, x, =, 10, ;, etc.)
2. **Parser:** Construye árbol sintáctico con 1 clase y 1 método
3. **Validator:** Verifica que tenga clase pública y método main
4. **Translator:** Genera código Python equivalente

Código Python de Salida:

```python
def main():
    x = 10
    print(x)

if __name__ == "__main__":
    main()
```

⚙️ Características Soportadas

☑ Elementos de Java Traducidos:

- ☑ Clases públicas
- ☑ Método `main`
- ☑ Variables (`int`, `double`, `boolean`, `String`, `char`)
- ☑ Operaciones matemáticas (`+`, `-`, `*`, `/`, `%`)
- ☑ Comparaciones (`==`, `!=`, `<`, `>`, `<=`, `>=`)
- ☑ Estructuras de control:
  - `if` / `else`

    - `while`
    - `for`
- ☑ `System.out.println()`
- ☑ Cadenas de texto y caracteres
- ☑ Comentarios (`//` y `/* */`)

❌ No Soportado (aún):

- ❌ Múltiples clases
- ❌ Herencia y polimorfismo
- ❌ Arrays complejos
- ❌ Excepciones (try/catch)
- ❌ Importaciones externas

🔘 Cómo Detener el Servidor

Opción 1: VS Code

1. Presiona `Ctrl + Shift + P`
2. Escribe "Run Task"
3. Selecciona "**🔘 Detener Servidor**"

Opción 2: Terminal

Presiona `Ctrl + C` en la terminal donde corre el servidor.

🩹 Solución de Problemas Comunes

Problema: "Cannot GET /"

Solución: Verifica que el servidor esté corriendo:

```
npm run server
```

Problema: "Puerto 3000 en uso"

Solución: Cambia el puerto en `src/server.js`:

```
const PORT = 3001; // Cambia a otro puerto
```

Problema: "Module not found"

Solución: Instala las dependencias:

```
npm install
```

Problema: El botón "Analizar" no responde

Solución:

1. Abre la consola del navegador (F12)
2. Verifica errores
3. Recarga la página (F5)

Archivos de Ejemplo

En la carpeta `src/ejemplos/` encontrarás:

- `test.java` - Programa básico de prueba
- `Main.java` - Clase simple con main

Puedes modificarlos o crear nuevos archivos `.java` para probar.

Consejos de Uso

1. **Escribe código simple:** El traductor funciona mejor con estructuras básicas
2. **Revisa los errores:** La pestaña de errores te guiará si algo falla
3. **Usa los ejemplos:** Carga ejemplos para aprender la sintaxis soportada
4. **Verifica el AST:** Si la traducción falla, revisa el árbol sintáctico
5. **Prueba el código Python:** Copia el resultado y ejecútalo en Python para verificar

---

Soporte

Si encuentras problemas:

1. Revisa la sección de **Solución de Problemas**
2. Verifica que todas las dependencias estén instaladas (`npm install`)
3. Consulta el **Manual Técnico** para más detalles

¡Disfruta traduciendo código!