



manual
USUARIO

The title is centered on the page. The word "manual" is written in a black, cursive, sans-serif font above the word "USUARIO". The word "USUARIO" is written in a large, bold, black, sans-serif font. A black wavy line is positioned below "USUARIO".

VALERY GALVEZ - 202200141

INTRODUCCION

El JavaLang Interpreter es una herramienta que permite escribir, ejecutar y depurar programas en el lenguaje académico JavaLang. Este manual explica paso a paso cómo instalar el sistema, crear y ejecutar programas con extensión .usl, y cómo interpretar los reportes generados (AST, tabla de símbolos y tabla de errores).

OBJETIVOS

- Objetivo General

Diseñar e implementar un intérprete para el lenguaje JavaLang, permitiendo la práctica de análisis léxico, sintáctico y semántico.

- Objetivos Específicos

- Desarrollar la gramática del lenguaje en Bison.
- Implementar un analizador léxico en Flex.
- Construir el Árbol de Sintaxis Abstracta (AST).
- Diseñar una interfaz gráfica con editor y consola.
- Generar reportes de tabla de símbolos, errores y AST.

INSTALACION

Requisitos Previos

- Sistema operativo: GNU/Linux Ubuntu 20.04 o superior.
- Paquetes necesarios:
sudo apt-get install build-essential flex bison libgtk-3-dev

Compilación

1. Descargar el proyecto desde el repositorio.
2. En la terminal, ubicarse en la carpeta del proyecto.
3. Ejecutar:
 - **make**
4. El ejecutable aparecerá en la carpeta bin/.

Si se hace desde la carpeta del proyecto, se debe hacer desde una terminal y se ejecuta lo siguiente:

```
unset LD_LIBRARY_PATH  
unset GTK_PATH  
unset GIO_MODULE_DIR  
make clean  
make ./javalang  
./javalang
```

USO DEL SISTEMA

Apertura de la Aplicación

- Ejecutar el programa

Creación y Edición de Archivos

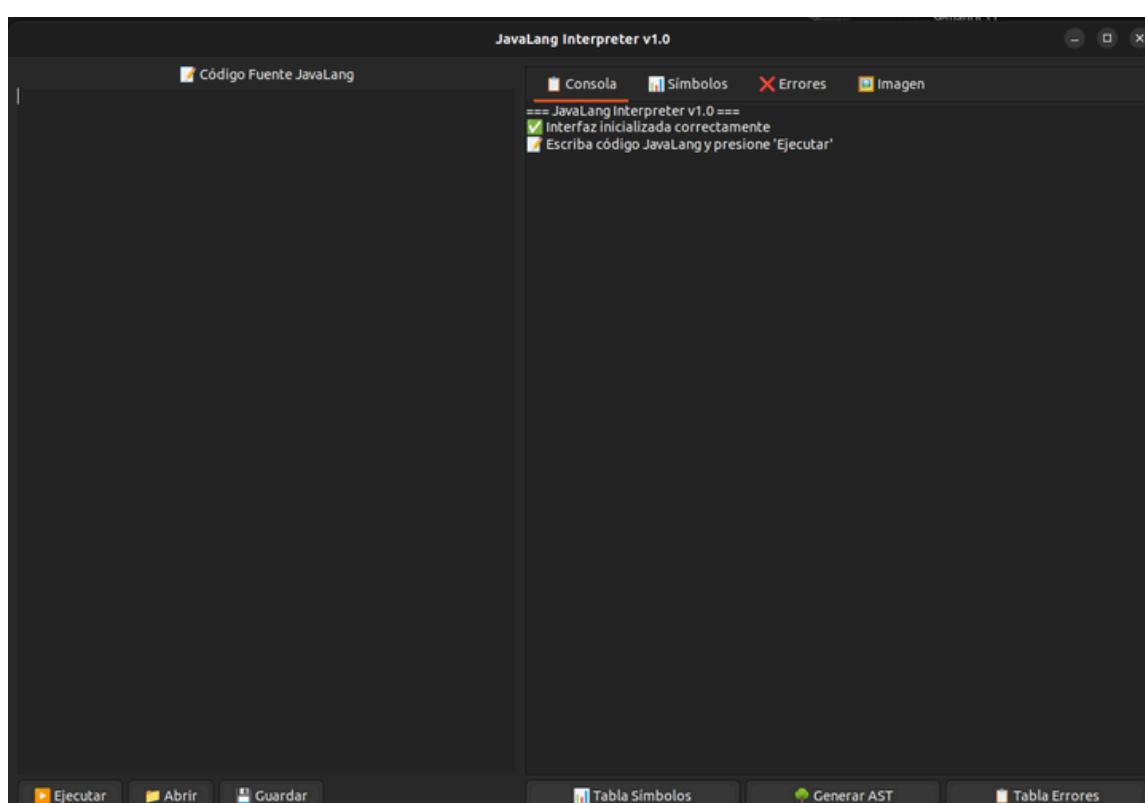
- En el menú Archivo → Nuevo, crear un archivo.
- Guardar con extensión .usl.
- Escribir el código en el editor integrado.

Ejecución de Programas

- Presionar el botón Ejecutar.
- La salida aparecerá en la consola integrada.

Reportes

- AST: muestra el árbol de sintaxis abstracta del programa.
- Tabla de símbolos: lista de variables y funciones declaradas.
- Errores: identifica errores léxicos, sintácticos o semánticos.



HERRAMIENTAS UTILIZADAS

- Lenguaje C (implementación base).
- Flex (análisis léxico).
- Bison (análisis sintáctico).
- Gtk3 (interfaz gráfica).
- Makefile (automatización de compilación).
- Sistema operativo recomendado: GNU/Linux Ubuntu.

EJEMPLOS DE USO

JavaLang Interpreter v1.0

Código Fuente JavaLang

```
public static void main() {
    // Operaciones mas complejas
    int a = 10;
    int b = 20;
    System.out.println("Suma: " + (a + b));
    System.out.println("Mayor: " + (a > b));

    // Condicionales
    if (a < b) {
        System.out.println("a es menor que b");
    }

    // Variables adicionales
    int resultado = a + b;
    System.out.println("Resultado guardado: " + resultado);

    // Operaciones booleanas
    boolean esMayor = a > b;
    System.out.println("a > b: " + esMayor);

    // Strings
    String mensaje = "Prueba completa";
    System.out.println("Mensaje: " + mensaje);
}
```

Consola Símbolos Errores Imagen

==== JavaLang Interpreter v1.0 ====
✓ Interfaz inicializada correctamente
Escriba código JavaLang y presione 'Ejecutar'

Iniciando ejecución...
✓ Análisis sintáctico: EXITOSO
Ejecutando programa...
Suma: 1020
Mayor: false
a es menor que b
Resultado guardado: 30
a > b: false
Mensaje: Prueba completa
✓ Análisis completado

Ejecutar Abrir Guardar Tabla Símbolos Generar AST Tabla Errores

JavaLang Interpreter v1.0

Código Fuente JavaLang

```
System.out.println("Condicion falsa, ejecutando else ");
puntosIfElse = puntosIfElse + 1;

System.out.println("\nIf-ElseIf-Else");
if (true) {
    System.out.println("Primera condicion verdadera");
} else if (true) {
    System.out.println("Segunda condicion verdadera, pero no se ejecuta");
} else {
    System.out.println("Ninguna condicion verdadera");
}

if (false) {
    System.out.println("Primera condicion falsa");
} else if (true) {
    System.out.println("Segunda condicion verdadera");
    puntosIfElse = puntosIfElse + 1;
} else {
    System.out.println("Ninguna condicion verdadera");
}

// 3. For Tipo While (2 puntos)
System.out.println("\n==== For Tipo While ====");
int puntosForWhile = 0;

System.out.println("For como while simple");
int i = 0;
int suma = 0;
while (i < 5) {
    System.out.println("i = " + i);
    suma = suma + i;
    i = i + 1;
}

if (suma == 10) {
    puntosForWhile = puntosForWhile + 1;
    System.out.println("OK For como while simple: correcto");
} else {
    System.out.println("X For como while simple: incorrecto");
}

System.out.println("\nFor como while anidado (patron X)");

```

Consola Símbolos Errores Imagen

===== Manejo de entornos =====
Variable redeclarada en el mismo entorno
a = 10
OK Detección de redeclaración en mismo entorno: correcto

Variable redeclarada en un entorno diferente
b dentro del if = 20
OK Redeclaración en entorno diferente: correcto
b fuera del if = 10

Uso de variable en un entorno superior
c dentro del if = 10
d dentro del if = 20
c fuera del if = 30
d fuera del if = 10
OK Uso de variable en entorno superior: correcto

===== If / Else =====
If simple
Condición verdadera

If-Else
Condición verdadera en if-else
Condición falsa, ejecutando else

If-ElseIf-Else
Primera condición verdadera
Segunda condición verdadera

===== For Tipo While =====
For como while simple
i = 0
i = 1
i = 2
i = 3
i = 4
OK For como while simple: correcto

For como while anidado (patrón X)
##Validación Manual

Ejecutar Abrir Guardar Tabla Símbolos Generar AST Tabla Errores

Tabla de Símbolos - JavaLang

31 Total Símbolos

| Nombre | Tipo | Ámbito | Línea |
|----------------|---------|--------|-------|
| pares | int | global | 0 |
| impares | int | global | 0 |
| puntosContinue | int | global | 0 |
| contador | int | global | 0 |
| puntosBreak | int | global | 0 |
| numeroBreak | int | global | 0 |
| numero | int | global | 0 |
| dia | int | global | 0 |
| puntosSwitch | int | global | 0 |
| valor | int | global | 0 |
| idx | int | global | 0 |
| sumalIndices | int | global | 0 |
| numeros | unknown | global | 0 |
| puntosForRange | int | global | 0 |
| jj | int | global | 0 |

JavaLang Interpreter v1.0

Símbolos TABLA DE SÍMBOLOS - JAVA LANG

ESTADÍSTICAS:

- Total de símbolos: 31
- Ámbito principal: Global (Nivel 0)

DETALLE DE SÍMBOLOS:

| No. | Nombre | Tipo | Ámbito | Línea |
|-----|------------------|---------|--------|-------|
| 1 | pares | int | global | 0 |
| 2 | impares | int | global | 0 |
| 3 | puntosContinue | int | global | 0 |
| 4 | contador | int | global | 0 |
| 5 | puntosBreak | int | global | 0 |
| 6 | numeroBreak | int | global | 0 |
| 7 | numero | int | global | 0 |
| 8 | dia | int | global | 0 |
| 9 | puntosSwitch | int | global | 0 |
| 10 | valor | int | global | 0 |
| 11 | idx | int | global | 0 |
| 12 | sumalIndices | int | global | 0 |
| 13 | numeros | unknown | global | 0 |
| 14 | puntosForRange | int | global | 0 |
| 15 | jj | int | global | 0 |
| 16 | puntosForClasico | int | global | 0 |
| 17 | j | int | global | 0 |
| 18 | x | int | global | 0 |
| 19 | n | int | global | 0 |
| 20 | suma | int | global | 0 |
| 21 | i | int | global | 0 |
| 22 | puntosForWhile | int | global | 0 |
| 23 | puntosIfElse | int | global | 0 |
| 24 | dLocal | int | global | 0 |
| 25 | d | int | global | 0 |
| 26 | c | int | global | 0 |
| 27 | bLocal | int | global | 0 |
| 28 | b | int | global | 0 |
| 29 | a | int | global | 0 |

Tabla Símbolos Generar AST Tabla Errores

Reporte de Errores - JavaLang

Tabla de Errores

Total: 1 errores encontrados

Léxicos: 0 | Sintácticos: 0 | Semánticos: 1 | ⚡ Runtime: 0

| No. | Tipo | Descripción | Línea | Columna | Ambito |
|-----|-----------|------------------------------------|-------|---------|--------|
| 1 | Semántico | Tipos incompatibles en comparación | 0 | 0 | Global |

Generado por JavaLang Interpreter - Sep 20 2025

JavaLang Interpreter v1.0

Errores REPORTE DETALLADO DE ERRORES

RESUMEN EJECUTIVO:

- Total de errores encontrados: 1
- Errores léxicos: 0
- Errores sintácticos: 0
- Errores semánticos: 1
- Errores de runtime: 0

DETALLES POR ERROR:

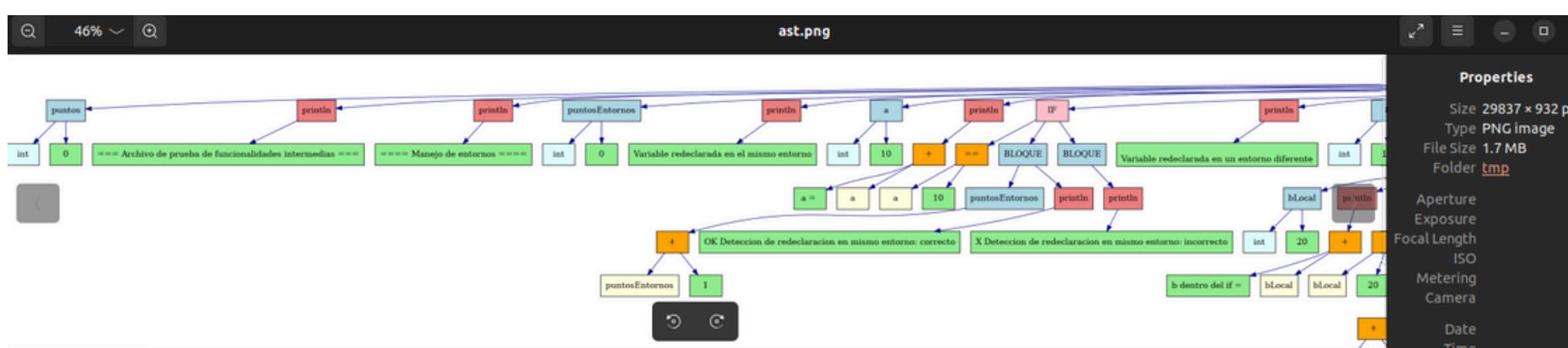
1. ERRORES SEMÁNTICO
Ubicación: Línea 0, Columna 0
Descripción: Tipos incompatibles en comparación
Ámbito: Global

SUGERENCIAS:

- Revisa los errores léxicos primero (caracteres inválidos)
- Corrige los errores sintácticos (estructura del código)
- Verifica las declaraciones y tipos de variables

JavaLang Interpreter v1.0

Tabla Símbolos Generar AST Tabla Errores



PROBLEMAS COMUNES

- Error: “command not found” → Verificar que se haya ejecutado make correctamente.
- Error de sintaxis inesperado → Revisar paréntesis o llaves mal cerradas en el código .usl.
- La GUI no abre → Confirmar que libgtk-3-dev esté instalado.
- Errores sin conocimiento de fallo → Usar macros de debug, para visualizar los errores mas comunes