October 24, 2025

Procesador de MyJS: *jsp*

Memoria del Grupo 59 (Primera Entrega)

Por Andrés Súnico (23M018)

Índice

1 Introducción

El desarrollo del procesador *jsp* se ha centrado en la experiencia del usuario (*UX*). Para ello, se han priorizado tres aspectos fundamentales una gestión de errores sólida e informativa, una interfaz de línea de comandos (*CLI*) intuitiva y, finalmente, que el programa sea eficiente y rápido.

Por estas razones se ha elegido *Rust* como el lenguaje de desarrollo. No sólo permite una gestión de memoria eficiente y segura, sino que también facilita el uso de *clap*, una de las mejores librerías para desarollar aplicaciones de *CLI*.

Además, gracias al uso del patrón de *inyección de dependencias* en todo el proyecto, el usuario puede optar por utilizar módulos (como el *Lexer* o la Tabla de Símbolos) que no vuelquen sus contenidos en ficheros, evitando así el *overhead* de las *syscalls*.

2 Información Adicional

El código fuente del procesador se puede encontrar en github.com/suuniqo, así como los tests y las dependencias del proyecto.

3 Opciones de la práctica

Además de las opciones comunes a todos los grupos, se han implementado las opciones:

3.1 Específicas del grupo

- Comentarios de bloque (/* */)
- Cadenas con comillas dobles (" ")
- Sentencia repetitiva do-while
- Asignación con *y* lógico (*&=*)
- Análisis Sintáctico Ascendente

3.2 Adicionales

Para que el procesador esté más completo, se han implementado adicionalmente todos los operadores lógicos, aritméticos, relacionales y unarios, así como los *tokens* booleanos *true* y *false*.

Además se ha escogido implementar el tratamiento de secuencias de escape (*\n* y *\t*).

4 Diseño del Lexer

El Analizador Léxico o *Lexer* es uno de los 3 módulos principales del procesador.

Al ser la primera capa de procesamiento, es el encargado de manejar el fichero fuente y convertirlo en una lista de *tokens* para el Analizador Sintáctico.

4.1 Tokens

Con el fin de lograr un procesamiento eficiente, tanto en memoria como en complejidad, se han minimizado el número de *tokens* con atributos.

De este modo sólo 4 de un total de 41 *tokens* van a utilizar un atributo.

Cabe notar, además, que se ha decidido no hacer uso del *token* fin de fichero (*EOF*). Esto es porque el *Lexer* se ha implementado como un iterador de *tokens*, de modo que el final del flujo se detecta naturalmente cuando se consume el iterador.

Table 1: Listado de *tokens*

Elemento	Código	Atributo
boolean	Bool	-
do	Do	-
float	Float	-
function	Func	-
if	If	-
int	Int	-
let	Let	-
read	Read	-
return	Ret	-
string	Str	-
void	Void	-
while	While	-
write	Write	-
constante real	FloatLit	Número
constante entera	IntLit	Número
Cadena	StrLit	Cadena
Identificador	Id	Posición
&=	AndAssign	-
=	Assign	-
,	Comma	-
;	Semi	-
(	LParen	-
)	RParen	-
{	LBrack	-
}	RBrack	-
Suma (+)	Sum	-
Por (*)	Mul	-
Resta (-)	Sub	-
División (/)	Div	-
Módulo (%)	Mod	-
Y lógico (&&)	And	-
O lógico (||)	Or	-
Negación (!)	Not	-
Menor (<)	Lt	-
Menor o igual (<=)	Le	-
Mayor (>)	Gt	-
Mayor o igual (>=)	Ge	-
Relacionales: Distinto (!=)	Ne	-
Igual (==)	Eq	-
Menos Unario (-)	Sub	-
Más Unario (+)	Sum	-
false	False	-
true	True	-

4.2 Errores

El diseño de los errores y del gestor está orientado a la *UX*. Cada tipo de error consta de un mensaje diferente y de una severidad, distinguiéndose *errors* de *warnings* (que no impedirían la compilación del programa).

El procesador genera mensajes claros con número de línea y columna, muestra la línea afectada y subraya en color (que depende de la severidad) la parte errónea.

El *Lexer* sólo genera un warning, *Invalid Escape Sequence*. Como se muestra en Acciones Semánticas, al detectar una secuencia de escape inválida no se descartara el *token* cadena, sino que se conserva literalmente (por ejemplo, la secuencia *\q*, se sustituye por esos dos mismos caracteres).

Table 2: Listado de errores del *Lexer*

Error	**Severidad**
Carácter inválido	*error*
Comentario inacabado	*error*
Cadena inacabada	error
Overflow de Cadena	*error*
Overflow de Entero	*error*
Overflow de Real	*error*
Formato de Real Inválido	*error*
Secuencia de Escape Inválida	*warning*

4.3 Gramática

Se define la gramática del *Lexer* como $G = (T, N, S, P)$, dónde:

$T = \{$Todo carácter *ASCII*$\} \cup \{$*EOF*$\}$
$N = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$

P se compone de las reglas:

$S \rightarrow$ del$S$ | , | ; | (|) | { | } | + | $*$ | % | $= A$ | $!B$ | $< C$ | $> D$ | $\&E$ | $|F$ | dG | $"H$ | $c_1 I$ | $/J$ | *EOF*
$A \rightarrow = \mid \lambda$
$B \rightarrow = \mid \lambda$
$C \rightarrow = \mid \lambda$
$D \rightarrow = \mid \lambda$
$E \rightarrow = \mid \lambda$
$F \rightarrow \mid$
$G \rightarrow dG \mid .K \mid \lambda$
$K \rightarrow dK \mid \lambda$
$H \rightarrow c_2 H \mid \backslash L \mid "$
$L \rightarrow$ n$H \mid$ tH
$I \rightarrow c_3 I \mid \lambda$
$J \rightarrow *M \mid \lambda$
$M \rightarrow c_4 M \mid *N$
$N \rightarrow c_5 M \mid *N \mid /S$

Y se definen:

$d := \{0, 1, \ldots, 9\}$
$l := \{$a, b, $\ldots$, z, A, B, $\ldots$, Z$\}$
$c_1 := l \cup \{\_\}$
$c_2 := T \setminus (\{\backslash, ", $*EOF*$\} \cup \{$Todos los caracteres *ASCII* no gráficos (Es decir, con código menor que 32)$\})$
$c_3 := c_1 \cup d$
$c_4 := T \setminus \{*, $*EOF*$\}$
$c_5 := T \setminus \{*, /, $*EOF*$\}$

4.4 Autómata

A continuación se muestra el autómata finito determinista que reconoce el lenguaje generado por la gramática G. Una transición "o.c." ocurre al leer un carácter que no corresponda a otra transición del estado.

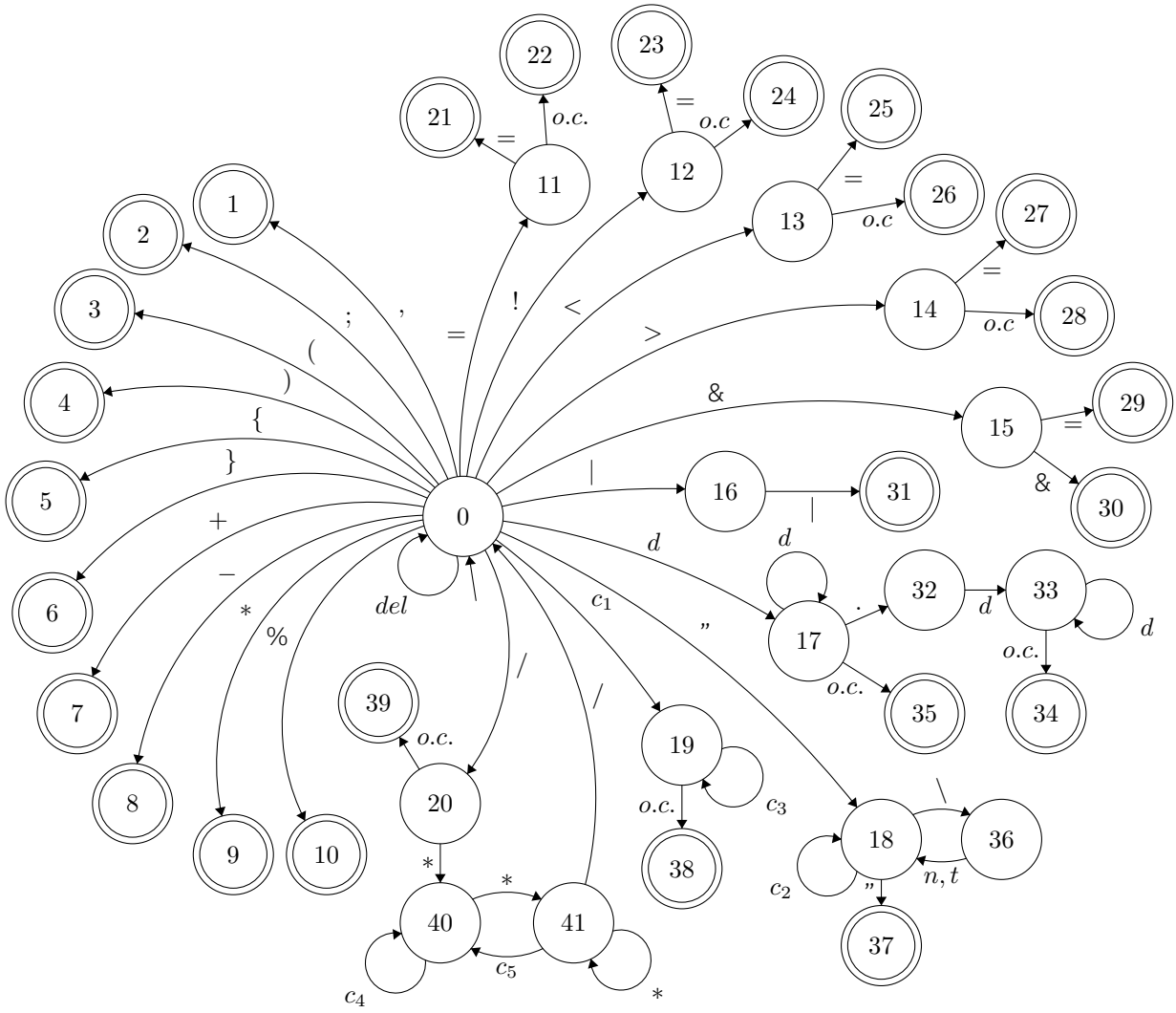

Figure 1: Autómata que reconoce el lenguaje $L(G)$

4.5 Acciones Semánticas

```
{
    "city": [
    {
        "id": 1,
        "name": "Toronto",
        "country": "Canada",
        "population": 6200000
    },
    {
        "id": 2,
        "name": "New York",
        "country": "United States of America",
        "population": 8800000
    }
    ]
}
```

5 Diseño de la Tabla de Símbolos

5.1 Estructura y Organización

6 Diseño del Parser

7 Diseño del Semanter

8 Gestor de Errores

Appendices

A Appendix Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget euismod pretium.

B Appendix Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget euismod pretium.

Aliquam arcu turpis, ultrices sed luctus ac, vehicula id metus. Morbi eu feugiat velit, et tempus augue. Proin ac mattis tortor. Donec tincidunt, ante rhoncus luctus semper, arcu lorem lobortis justo, nec convallis ante quam quis lectus. Aenean tincidunt sodales massa, et hendrerit tellus mattis ac. Sed non pretium nibh. Donec cursus maximus luctus. Vivamus lobortis eros et massa porta porttitor.

Fusce varius orci ac magna dapibus porttitor. In tempor leo a neque bibendum sollicitudin. Nulla pretium fermentum nisi, eget sodales magna facilisis eu. Praesent aliquet nulla ut bibendum lacinia. Donec vel mauris vulputate, commodo ligula ut, egestas orci. Suspendisse commodo odio sed hendrerit lobortis. Donec finibus eros erat, vel ornare enim mattis et. Donec finibus dolor quis dolor tempus consequat. Mauris fringilla dui id libero egestas, ut mattis neque ornare. Ut condimentum urna pharetra ipsum consequat, eu interdum elit cursus. Vivamus scelerisque tortor et nunc ultricies, id tincidunt libero pharetra. Aliquam eu imperdiet leo. Morbi a massa volutpat velit condimentum convallis et facilisis dolor.

C Appendix Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget euismod pretium.