

Manual Técnico

Minimal es un proyecto que consta de dos aplicaciones: un editor de texto para el usuario y un servidor de procesamiento. Su objetivo es proporcionar un entorno donde los usuarios puedan escribir y analizar contenido, con soporte para WebSockets y procesamiento de lenguajes mediante analizadores léxicos y sintácticos.

Tecnología Usadas

Java 21: Lenguaje de programación principal del proyecto



Java Swing: Para la interfaz gráfica del editor de usuario

Spring Boot: Para el desarrollo del servidor y la gestión de WebSockets



JFlex y CUP: Herramientas para la creación de analizadores
léxicos y sintácticos

The JFlex logo consists of the word "JFlex" in a white, sans-serif font, centered within a white square border.

JFlex

The JavaCup logo features the word "JavaCup" in a stylized, italicized font. The "C" is large and the "u" is small. A coffee cup icon is integrated into the design, replacing the dot of the "i" in "Cup". The logo is set against a light gray rectangular background.

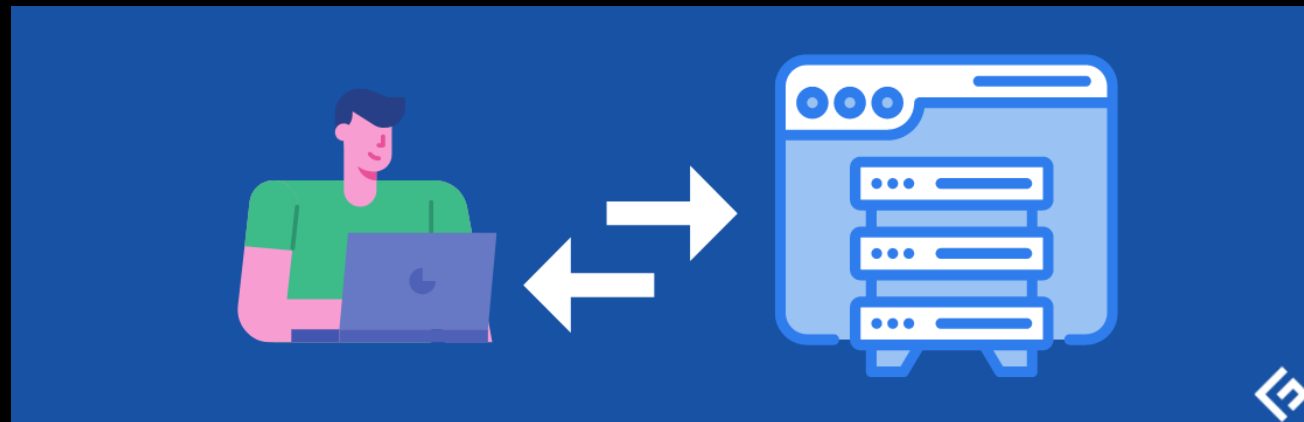
JavaCup

ANTLR4: Para el manejo avanzado de gramáticas y parsing de lenguajes



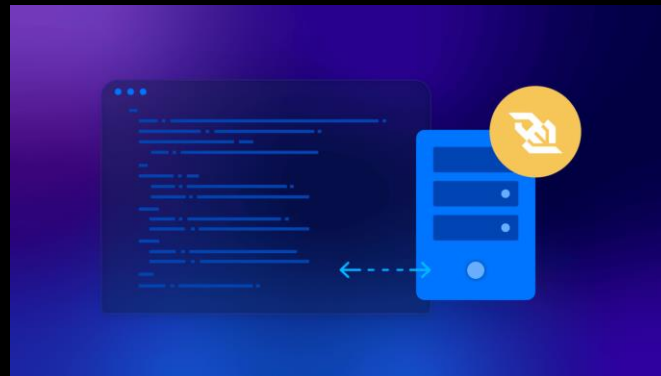
Editor de Usuario (Cliente)

- Implementado en **Java Swing**.
- Permite a los usuarios escribir y editar contenido.
- Se comunica con el servidor mediante **WebSockets**.



Servidor

- Implementado en Spring Boot.
- Maneja las conexiones WebSocket para la comunicación en tiempo real.
- Utiliza JFlex y CUP para la generación y ejecución de analizadores léxicos y sintácticos.
- ANTLR4 se usa para el parsing de estructuras más complejas.



Instalación y Configuración

- NetBeans Instalado.
- JDK 21 configurado.
- Dependencias necesarias en nave pom.xml

```
<!-- DEPENDENCIA DE WEBSOCKET -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-websocket</artifactId>
</dependency>
<!-- DEPENDENCIA CUP RUN TIME -->
<dependency>
    <groupId>com.github.vbmacher</groupId>
    <artifactId>java-cup-runtime</artifactId>
    <version>11b-20160615</version>
</dependency>
<!-- DEPENDENCIA DE PICOCLI -->
<dependency>
    <groupId>info.picocli</groupId>
    <artifactId>picocli</artifactId>
    <version>4.7.6</version>
</dependency>
<!-- DEPENDENCIA DE ANTLR4 -->
<dependency>
    <groupId>org.antlr</groupId>
    <artifactId>antlr4</artifactId>
    <version>4.13.2</version>
</dependency>
```


Gramatica Minimal jflex y cup

```
INCRE="++"  
DECRE="--"  
POT="^"  
MAS="+"  
MENOS="-"  
POR="*"  
DIV="/"  
PAR1="("  
PAR2=")"  
IGUAL="="  
MENOR="<"  
MAYOR=">"  
OR="|"  
AND="&&"  
DIFER="!"  
FINCADENA=";"  
LLAVE1="{ "  
LLAVE2="} "  
DOSPTS=":"  
COMA=","  
PUNTO="."
```

```
CONST="const"  
RETURN="return"  
BREAK="break"  
NUMBER="number"  
STRING="string"  
CHAR="char"  
BOOLEAN="boolean"  
VOID="void"  
VAR="var"  
CONSOLE="console"  
LOG="log"  
FUNCTION="function"  
IF="if"  
ELSE="else"  
FOR="for"  
TRUE="true"  
FALSE="false"
```

```
MAIN="main"  
P="p"  
BUTTON="button"  
HNUM="h"[1-6]  
TAG_OPEN={MENOR}{HNUM}{MAYOR}  
TAG_CLOSE={MENOR}{DIV}{HNUM}{MAYOR}  
PARRAFO_OPEN={MENOR}{P}{MAYOR}  
PARRAFO_CLOSE={MENOR}{DIV}{P}{MAYOR}  
BUTTON_OPEN={MENOR}{BUTTON} ([^>])* {MAYOR}  
BUTTON_CLOSE={MENOR}{DIV}{BUTTON}{MAYOR}  
INPUT="input"  
VALUE="value"  
ONCLICK="onClick"
```

```
BLANCOS=[\ \r\t\f\n]+  
ENTERO=[0-9]+  
DECIMAL=[0-9]+ "." [0-9]+  
IDENTIFICADOR=[a-zA-Z][a-zA-Z0-9_]*  
CADENA=[\"]([^\"])*[\" ]  
CARACTER = [\\']([\\'\\\"])*[\\']
```

```
terminal INCRE, DECRE;
terminal POT, MAS, MENOS, POR, DIV, UMENOS;
terminal PAR1, PAR2, IGUAL, MENOR, MAYOR, EQUALS;
terminal OR, AND, DIFER, NOT, FINCADENA;
terminal LLAVE1, LLAVE2, DOSPTS, COMA, PUNTO;
terminal CONST, RETURN, NUMBER, STRING, CHAR, BREAK;
terminal BOOLEAN, VOID, VAR, CONSOLE, LOG;
terminal FUNCTION, IF, ELSE, FOR, TRUE, FALSE;
terminal String TAG_OPEN, TAG_CLOSE;
terminal String PARRAFO_OPEN, PARRAFO_CLOSE;
terminal String BUTTON_OPEN, BUTTON_CLOSE;
terminal MAIN, INPUT, VALUE, ONCLICK;
terminal String ENTERO, DECIMAL, IDENTIFICADOR, CADENA, CARACTER;
terminal String HTML_CONTENT;
```

```
precedence left AND, OR, DIFER;
precedence left MENOR, MAYOR;
precedence left MAS, MENOS;
precedence left POR, DIV;
precedence right POT;
precedence left UMENOS;
precedence left EQUALS;
precedence left NOT;
```

```
non terminal LinkedList<Instruccion> INICIO, INSTRUCCIONES, CONTENIDOS;
non terminal Instruccion INSTRUCCION, CONTENIDO;
non terminal Instruccion IMPRESION, DECLARACION, ASIGNACION;
non terminal Instruccion SIF, BREAKK, RETU;
non terminal Instruccion EXPRESION;
non terminal Instruccion CFOR;
non terminal Instruccion FUNCION, LLAMADA_FUNCION;
non terminal Instruccion ETIQUETAS, PARRAFOS, INPUTS, BUTTONS;
non terminal Instruccion DATO, VARIABLE;
non terminal LinkedList<Parametro> PARAMETROS;
non terminal Parametro PARAMETRO;
non terminal Tipo TIPOS;
non terminal MAIN1, MAIN2;
```

```
INICIO -> CONST IDENTIFICADOR IGUAL PAR1 PAR2 IGUAL MAYOR LLAVE1
        INSTRUCCIONES RETURN PAR1 MAIN1 CONTENIDOS MAIN2 PAR2 FINCADENA LLAVE2
```

```
MAIN1 -> MENOR MAIN MAYOR
```

```
MAIN2 -> MENOR DIV MAIN MAYOR
```

```
INSTRUCCIONES -> INSTRUCCION
                | INSTRUCCIONES INSTRUCCION
```

```
INSTRUCCION -> IMPRESION
                | DECLARACION
                | ASIGNACION
                | SIF
                | BREAKK
                | RETU
                | CFOR
                | FUNCION
                | LLAMADA_FUNCION
```

```
IMPRESION -> CONSOLE PUNTO LOG PAR1 EXPRESION PAR2 FINCADENA
```

```
DECLARACION -> VAR IDENTIFICADOR DOSPTS TIPOS IGUAL EXPRESION FINCADENA
```

```
ASIGNACION -> IDENTIFICADOR IGUAL EXPRESION FINCADENA
                | IDENTIFICADOR INCRE
                | IDENTIFICADOR INCRE FINCADENA
                | IDENTIFICADOR DECRE
                | IDENTIFICADOR DECRE FINCADENA
```

```
SIF -> IF PAR1 EXPRESION PAR2 LLAVE1 INSTRUCCIONES LLAVE2
        | IF PAR1 EXPRESION PAR2 LLAVE1 INSTRUCCIONES LLAVE2 ELSE SIF
        | IF PAR1 EXPRESION PAR2 LLAVE1 INSTRUCCIONES LLAVE2 ELSE LLAVE1 INSTRUCCIONES LLAVE2
```

```
BREAKK -> BREAK FINCADENA
```

```
RETU -> RETURN EXPRESION FINCADENA
```

```
CFOR -> FOR PAR1 DECLARACION EXPRESION FINCADENA ASIGNACION PAR2 LLAVE1 INSTRUCCIONES LLAVE2
```

```
FUNCION -> FUNCTION IDENTIFICADOR PAR1 PARAMETROS PAR2 DOSPTS TIPOS LLAVE1 INSTRUCCIONES LLAVE2
```

```
PARAMETROS -> PARAMETRO
                | PARAMETROS COMA PARAMETRO
                | e
```

```
PARAMETRO -> IDENTIFICADOR DOSPTS TIPOS
```

```
LLAMADA_FUNCION -> IDENTIFICADOR PAR1 PARAMETROS PAR2 FINCADENA
```

```
TIPOS -> NUMBER
        | STRING
        | CHAR
        | BOOLEAN
        | VOID
```

```

EXPRESION  -> MENOS EXPRESION
            | DIFER EXPRESION
            | EXPRESION MAS EXPRESION
            | EXPRESION MENOS EXPRESION
            | EXPRESION POR EXPRESION
            | EXPRESION DIV EXPRESION
            | EXPRESION POT EXPRESION
            | EXPRESION MENOR IGUAL EXPRESION
            | EXPRESION MAYOR IGUAL EXPRESION
            | EXPRESION MENOR EXPRESION
            | EXPRESION MAYOR EXPRESION
            | EXPRESION OR EXPRESION
            | EXPRESION AND EXPRESION
            | ENTERO
            | DECIMAL
            | CADENA
            | CARACTER
            | TRUE
            | FALSE
            | IDENTIFICADOR
            | PAR1 EXPRESION

```

```

CONTENIDOS -> CONTENIDO
            | CONTENIDOS CONTENIDO

```

```

CONTENIDO  -> ETIQUETAS
            | PARRAFOS
            | INPUTS
            | BUTTONS

```

```

ETIQUETAS  -> TAG_OPEN DATO TAG_CLOSE

```

```

PARRAFOS   -> PARRAFO_OPEN DATO PARRAFO_CLOSE

```

```

DATO        -> HTML_CONTENT

```

```

VARIABLE    -> LLAVE1 IDENTIFICADOR LLAVE2

```

```

INPUTS      -> MENOR INPUT VALUE IGUAL VARIABLE DIV MAYOR

```

```

BUTTONS     -> BUTTON_OPEN DATO BUTTON_CLOSE

```

Gramatica shttp scl antlr4



```
1 COMA : ',';
2 TREPTS : '...';
3 GET : 'GET';
4 POST : 'POST';
5 PATCH : 'PATCH';
6 DELETE : 'DELETE';
7 SITIO : 'SITIO';
8 PAGINA : 'PAGINA';
9 MSITIO : 'sitio';
10 MPAGINA : 'pagina';
11 ABRIR : 'abrir';
12 CREAR : 'crear';
13 AGREGAR : 'agregar';
14 ELIMINAR : 'eliminar';
15 MODIFICAR : 'modificar';
16 IDENTIFICADOR : [a-zA-Z][a-zA-Z0-9_-]*;
17 BLANCOS : [ \t\r\n]+ -> skip;
18 CONTENIDO : '<main>' .*? '</main>';
```



```
1 init: instrucciones;
2
3 instrucciones: instruccion+;
4
5 instruccion: get
6             | post
7             | patch
8             | delete
9 ;
10
11 get: GET (SITIO | PAGINA) ABRIR (MSITIO | MPAGINA) parametros;
12
13 post: POST (SITIO | PAGINA) CREAR (MSITIO | MPAGINA) parametros body?;
14
15 patch: PATCH (SITIO | PAGINA) (MODIFICAR | AGREGAR) (MSITIO | MPAGINA) parametros body?;
16
17 delete: DELETE (SITIO | PAGINA) ELIMINAR (MSITIO | MPAGINA) parametros;
18
19 parametros: parametro (COMA parametro)*;
20
21 parametro: IDENTIFICADOR;
22
23 body: TREPTS CONTENIDO TREPTS;
```

Gramatica toml antlr4



```
1 PUNTO : '.';
2 IGUAL : '=';
3 COR1 : '[';
4 COR2 : ']';
5 NOMBRE : 'nombre';
6 PATH : 'path';
7 IDENTIFICADOR : [a-zA-Z][a-zA-Z0-9_]*;
8 ATRIBUTO : '"' (~["\r\n])* '"';
9 COMENTARIO : '#' ~[\r\n]* -> skip;
10 BLANCOS: [\t\r\n]+ -> skip;
```



```
1 init: instrucciones;
2
3 instrucciones: instruccion+;
4
5 instruccion: etiquetas
6             | atributos
7 ;
8
9 etiquetas: COR1 valores COR2;
10
11 valores: valor (PUNTO valor)*;
12
13 valor: IDENTIFICADOR;
14
15 atributos: (NOMBRE | PATH) IGUAL ATRIBUTO;
```

Diagrama de Cliente



Diagrama de Servidor

Servidor

