

Manual Tecnico

Autor: James Osmin Gramajo Cárcamo 201731172

Requerimientos Mínimos:

- Procesador: 2 GHz compatible con PAE, NX y SSE2.
- RAM: 1 GB (32 bits) o 2 GB (64 bits).
- Espacio en disco duro: 16 GB (32 bits) o 20 GB (64 bits).
- Tarjeta gráfica: Dispositivo gráfico Microsoft DirectX 9 con controlador WDDM
- Resolución de pantalla de al menos 1366 x 768 píxeles.
- Es posible que algunos juegos y programas requieran tarjetas gráficas compatibles con DirectX 10 o superior para un rendimiento óptimo.
- JDK 11
- Maquina virtual(Virtual machine) o cualquier otro sistemas operativo.
- Windows o Linux.
- jison
- Angular
- Nodejs
- paqueteria npm

Alojamiento del código fuente del proyecto en el repositorio GitHub:

<https://github.com/jamesg19/Practica2>

El módulo de Node no fue posible cargarlo al repositorio por lo tanto dejo el link para agregarlo al proyecto:

<https://docs.google.com/document/d/1t6FnVZcwfgbqPnTtyGszjgUMtFXPILCMsulPYvzJZH4/edit?usp=sharing>

Gramatica generada para el reconocimiento de Gramatica Lexica Wison

Reconocimiento lexico y expresiones utilizadas:

```
lexic [lL][eE][xX]
termin [tT][eE][rR][mM][iI][nN][aA][lL]
nterm [nN][oO]"_"[tT][eE][rR][mM][iI][nN][aA][lL]
wiss [W][iI][sS][oO][nN]
syntaxis [sS][yY][nN][tT][aA][xX]
initialSIM [iI][nN][iI][tT][iI][aA][lL]"_"[sS][iI][mM]
int "-"?([0-9][1-9][0-9]+)
exp [eE][-+]?[0-9]+
frac "."[0-9]+
letras [a-zA-Z]+
espacio [\s\\n\\t\\r]+
identificador [a-zA-Z0-9]+([a-zA-Z][0-9]+)*
especial [á-ü][«#%&/()=*.+-]
{wiss}          return 'WISON'
"¿"             return 'QUESTIONMARK_L'
{lexic}         return 'LEXX'
"{:"           return 'INICIO_LEXICO'
{termin}        return 'TERMINAL'
{nterm}         return 'NO_TERMINAL'
{syntaxis}      return 'SYNTAX_INI'
{initialSIM}    return 'INITIAL_SYM'
"{{:"          return 'INICIO_SYNTAX'
"<_"           return 'MENORQ_MENOS'
"<="          return 'MENORQ_IGUAL'
";"            return 'PUNTO_COMA'
"|"           return 'O'
"$_{(identificador)}_" return 'NOMBRE_TERMINAL'
"%_{(identificador)}_" return 'NOMBRE_NOTERMINAL'
("'|"')        this.begin("string");
<string>[^"]+  return "CHARACTER";
<string>("'|"') this.popState();
"[aA-zZ]"      return 'TODAS'
"[0-9]"        return 'NUMEROS'
"*"           return 'ESTRELLA'
"+"           return 'CERRADURA'
"?"           return 'QUESTIONMARK_D'
":.}"         return 'FIN_SYNTAX'
":.}"         return 'FIN_LEXICO'
{int}{frac}?b return 'NUMBER'
{letras}      return 'LETRAS'
("#")         this.begin("coment");
<coment>[^\\n\\r]+ //return "COMENTARIO";
<coment>[\\n]    this.popState();
```

```

{espacio}           //ignora el espacio en blanco

[/][*][^]*[*]+([/*][^]*[*]+)*[/]      { /*ignorar comentario de bloque*/}
<<EOF>>           return 'EOF'
.                 return 'INVALID'

```

Gramatica generada para el reconocimiento de Gramatica Sintactica Wison

```

ini: analizador EOF
;
analizador :
    WISON QUESTIONMARK_L
    LEXX INICIO_LEXICO
    terminales
    FIN_LEXICO
    SYNTAX_INI INICIO_SYNTAX
    no_terminales
    bloque_producciones
    FIN_SYNTAX
    QUESTIONMARK_D WISON
    | error {
        console.log('Este es un error: '+yytext+
            ', en la linea: '+ (this._$.first_line+1)+
            ', en la columna: '+ (this._$.first_column+1));
    }
;
terminales:
    TERMINAL NOMBRE_TERMINAL MENORQ_MENOS caracteres PUNTO_COMA
terminales
    | TERMINAL NOMBRE_TERMINAL MENORQ_MENOS caracteres PUNTO_COMA
;
caracteres:
    TODAS
    | NUMEROS
    | CHARACTER
    | identificador
    | TODAS ejecucion
    | NUMEROS ejecucion
    | CHARACTER ejecucion
    | identificador ejecucion
;
ejecucion:
    ESTRELLA
    | CERRADURA
    | QUESTIONMARK_D
;
//DEFINO LA GRAMTICA DEL SINTACTICO DE WISON
no_terminales:

```

```

    NO_TERMINAL NOMBRE_NOTERMINAL PUNTO_COMA no_terminales
    | NO_TERMINAL NOMBRE_NOTERMINAL PUNTO_COMA
;
bloque_producciones:
    INITIAL_SYM NOMBRE_NOTERMINAL PUNTO_COMA
    producciones
;
producciones:
    NOMBRE_NOTERMINAL MENORQ_IGUAL gramatica PUNTO_COMA producciones
    | NOMBRE_NOTERMINAL MENORQ_IGUAL gramatica PUNTO_COMA
;
gramatica:
    NOMBRE_NOTERMINAL gramatica
    | NOMBRE_TERMINAL gramatica
    | NOMBRE_NOTERMINAL
    | NOMBRE_TERMINAL
;

```

Comandos utilizados para ejecutar la aplicacion

ng serve

el siguiente comando es utilizado en consola y dentro de la carpeta que contiene el proyecto
ejemplo

C:/Users/james/Documents/Practica:

se debera ejecutar dicho comando

C:/Users/james/Documents/Practica: ng serve

EL modulo de de angular no es posible agregarlo al repositorio