

## Manual Técnico

A continuación, se mostrará las partes del analizador léxico y sintáctico que fueron utilizados para la realización del análisis del proyecto:

### Analisis Lexico:

Como primer punto se tomaron los valores que no iban a ser parte del análisis como son los saltos de línea o regresión del carrete, así como los espacios en blanco y las tabulaciones.

Luego de esto se procedió a realizar la declaración de las expresiones regulares que serían utilizadas para la validación de cadenas y de números enteros.

Por último, se declaró la acción a realizar cuando se encontrará ya sea las palabras o dígitos o las palabras clave y signos, en este caso se retorna un elemento Symbol que va servir más adelante para el análisis sintáctico y luego de esto se especificó que cualquier otro error que se encontrará en relación con el alfabeto fuera notificado por excepción la cual ya que manejar.

`Salto = \r|\n|\r\n`

`Espacios = {Saltos} | [ \t\f]`

`Palabra = [a-zA-Z][a-zA-Z]*[0-9]*`

`Digito = [0-9][0-9]*`

`%% // Separador de Area`

`<YYINITIAL> {`

`SELECCIONAR { return symbol(sym.SELECCIONAR); }`

`INSERTAR { return symbol(sym.INSERTAR); }`

`ACTUALIZAR { return symbol(sym.ACTUALIZAR); }`

`ELIMINAR { return symbol(sym.ELIMINAR); }`

`EN { return symbol(sym.EN); }`

`FILTRAR { return symbol(sym.FILTRAR); }`

`VALORES { return symbol(sym.VALORES); }`

`ASIGNAR { return symbol(sym.ASIGNAR); }`

`"\" { return symbol(sym.COMILLA); }`

```

"("    { return symbol(sym.PARANTESIS_I); }

")"    { return symbol(sym.PARANTESIS_D); }

"."    { return symbol(sym.PUNTO); }

","    { return symbol(sym.COMA); }

";"    { return symbol(sym.PUNTO_Y_COMA); }

"*"    { return symbol(sym.POR); }

"="    { return symbol(sym.IGUAL); }

"<"    { return symbol(sym.MENOR); }

">"    { return symbol(sym.MAYOR); }

"<="   { return symbol(sym.MENOR_IGUAL); }

">="   { return symbol(sym.MAYOR_IGUAL); }

{Palabra}    { return symbol(sym.PALABRA, yytext()); }

{Digito} { return symbol(sym.ENTERO, yytext()); }

{Espacios}  { /* ignoramos */ }

}

[^]    { throw new Error("Error Léxico caracter Invalido en la linea " + yyline + ", columna " +
yycolumn + ": " + yytext()); }

```

### Analisis Sintactico:

Para este análisis se pensó como primer punto en que los valores iniciales deberían ser las palabras clave SELECCIONAR, INSERTAR, ACTUALIZAR Y ELIMINAR, luego de esto se tuvo que dividir cada análisis en su producción similares pero las cuales no podían ir juntas ya que nos podría dar una combinación de instrucciones que al momento de ejecutarlas no tuvieran ningún sentido, por lo tanto, se dividió.

Por otro lado, se declararon los valores terminarles siguiendo la Palabra y el Entero los únicos valores declarados como String y int respectivamente.

Y para el manejo de errores se utilizó el `syntax_error()` el cual que se utiliza cuando la secuencia de token no concide con la gramatica establecida.

```
/* Gramatica */
```

```
start with inicial;
```

```
inicial ::= SELECCIONAR seleccion
```

```
    | INSERTAR EN raiz2
```

```
    | ACTUALIZAR EN raiz3
```

```
    | ELIMINAR EN raiz4
```

```
;
```

```
/* Seleccionar */
```

```
seleccion ::= PALABRA COMA columna
```

```
    | PALABRA EN raiz
```

```
    | POR EN raiz
```

```
;
```

```
columna ::= PALABRA COMA columna
```

```
    | PALABRA EN raiz
```

```
;
```

```
raiz ::= PALABRA PUNTO proyecto
```

```
;
```

```
proyecto ::= PALABRA PUNTO proyecto
```

```
    | PALABRA FILTRAR filtro
```

```
;
```

```
/* Insertar */
```

```
raiz2 ::= PALABRA PUNTO proyecto2
```

```
;
```

```
proyecto2 ::= PALABRA PUNTO proyecto
          | PALABRA PARANTESIS_I columna2
          | PALABRA POR VALORES PARANTESIS_I valores
          ;
```

```
columna2 ::= PALABRA COMA columna2
          | PALABRA PARANTESIS_D VALORES PARANTESIS_I valores
          ;
```

```
valores ::= COMILLA PALABRA COMILLA COMA valores
          | ENTERO COMA valores
          | COMILLA PALABRA COMILLA PARANTESIS_D PUNTO_Y_COMA inicial
          | ENTERO PARANTESIS_D PUNTO_Y_COMA inicial
          | COMILLA PALABRA COMILLA PARANTESIS_D PUNTO_Y_COMA
          | ENTERO PARANTESIS_D PUNTO_Y_COMA
          ;
```

```
/* Actualizar */
```

```
raiz3 ::= PALABRA PUNTO proyecto3
      ;
```

```
proyecto3 ::= PALABRA PUNTO proyecto3
            | PALABRA ASIGNAR columna3
            ;
```

```
columna3 ::= PALABRA IGUAL COMILLA PALABRA COMILLA COMA columna3
            | PALABRA IGUAL ENTERO COMA columna3
            | PALABRA IGUAL COMILLA PALABRA COMILLA FILTRAR filtro
            | PALABRA IGUAL ENTERO FILTRAR filtro
            | PALABRA IGUAL COMILLA PALABRA COMILLA PUNTO_Y_COMA inicial
            | PALABRA IGUAL ENTERO PUNTO_Y_COMA inicial
```

```
| PALABRA IGUAL COMILLA PALABRA COMILLA PUNTO_Y_COMA
| PALABRA IGUAL ENTERO PUNTO_Y_COMA
;
```

```
/* Eliminar */
```

```
raiz4 ::= PALABRA PUNTO proyecto4
```

```
;
```

```
proyecto4 ::= PALABRA PUNTO proyecto4
```

```
| PALABRA FILTRAR filtro
| PALABRA PUNTO_Y_COMA inicial
| PALABRA PUNTO_Y_COMA
;
```

```
filtro ::= PALABRA IGUAL complemento1
```

```
| PALABRA MENOR complemento2
| PALABRA MAYOR complemento2
| PALABRA MENOR_IGUAL complemento2
| PALABRA MAYOR_IGUAL complemento2
;
```

```
complemento1 ::= COMILLA PALABRA COMILLA PUNTO_Y_COMA inicial
```

```
| ENTERO PUNTO_Y_COMA inicial
| COMILLA PALABRA COMILLA PUNTO_Y_COMA
| ENTERO PUNTO_Y_COMA
;
```

```
complemento2 ::= ENTERO PUNTO_Y_COMA inicial
```

```
| ENTERO PUNTO_Y_COMA
;
```