

# MANUAL TÉCNICO:

## EMULADOR DB:

El presente programa consiste en un emulador de instrucciones de SQL, el cual tiene la funcionalidad de ejecutar acciones como lo haría un motor de base de datos real.

A continuación, se presenta el análisis léxico y la gramática en formato BNF.

## Análisis léxico:

### Expresiones regulares:

```
numero = [0-9]
entero = [0-9]+
decimal = [0-9]+(\.[0-9]+)?
letra = [A-Za-z]
simbolo = [\_\@+\*\#]
nombreArchivo = ({numero} | {simbolo} | {letra})+
//([\_ \@+\*\#] | [a-zA-Z0-9])+([\_ \@+\*\#] | [a-zA-Z0-9])*

identificador = [a-zA-Z]+([0-9a-zA-Z])*
cadena="\[^"]*"

```

**Signos a identificar:** >, <, >=, <=, (, ), <>

**Palabras reservadas:** AND, OR, INSERTAR, SELECCIONAR, ACTUALIZAR, ELIMINAR, FILTRAR, VALORES, EN, ASIGNAR.

**Otros símbolos:** \*, coma (,), punto y coma (;)

## Análisis Sintáctico:

**Gramática:** para la siguiente práctica se uso la siguiente gramática, incluye los terminales y no terminales. La gramática está definida según la sintaxis de JCUP.

```
terminal String      SELECCIONAR, INSERTAR, ELIMINAR, ACTUALIZAR,
terminal String      FILTRAR, EN, AND, OR;
terminal String      CADENA, ID, ARCHIVO, VALORES, ASIGNAR;
terminal int         ENTERO;
```

```
terminal double    DECIMAL;
terminal String    SIG_IGUAL,  SIG_MENOR,  SIG_MAYOR,
SIG_MENOR_QUE, SIG_MAYOR_QUE;
terminal String    SIG_DIFERENTE, PCOMA, PUNTO, COMA, ASTERISCO,
PARENA, PARENC;
```

```
/* Non-terminals */
```

```
non terminal      instruccion, select, insert, update, delete,
lista_columnas, ruta_archivo;
```

```
non terminal      condiciones, value_list, next_file, and_or, numero,
signos, and, or;
```

```
non terminal      op_select, lista_valores, valores, end_lista_valores,
update_list, end_up_list;
```

```
/* The grammar */
```

```
start with instruccion;
```

```
instruccion  ::= select PCOMA
              | insert PCOMA
              | update PCOMA
              | delete PCOMA
              ;
```

```
//-----> seleccionar
select      ::= SELECCIONAR op_select
              ;
```

```
op_select   ::= lista_columnas EN ruta_archivo condiciones
              | ASTERISCO EN ruta_archivo condiciones
              ;
```

```
//-----> insertar
insert      ::= INSERTAR EN ruta_archivo PARENA lista_columnas
PARENC VALORES PARENA lista_valores PARENC
              | INSERTAR EN ruta_archivo VALORES PARENA lista_valores
PARENC
              ;
```

```
//-----> actualizar
update      ::= ACTUALIZAR EN ruta_archivo ASIGNAR update_list
condiciones;
```

```
update_list ::= ID SIG_IGUAL valores end_up_list;
```

```
end_up_list ::= COMA update_list
```

```

        |
        ;

//-----> eliminar
delete      ::= ELIMINAR EN ruta_archivo condiciones;

lista_columnas ::= ID value_list;

value_list  ::= COMA lista_columnas
              | /* Nada */
              ;

ruta_archivo ::= ID next_file;

next_file   ::= PUNTO ruta_archivo
              | /* Nada */
              ;

condiciones ::= FILTRAR ID SIG_IGUAL CADENA and_or
              | FILTRAR ID signos numero and_or
              | /* Nada */
              ;

and_or      ::= and
              | or
              ;

and         ::= AND ID SIG_IGUAL CADENA and
              | AND ID signos numero and
              ;

or          ::= OR ID SIG_IGUAL CADENA or
              | OR ID signos numero or
              ;

signos      ::= SIG_MAYOR
              | SIG_MAYOR_QUE
              | SIG_MENOR
              | SIG_MENOR_QUE
              | SIG_DIFERENTE
              | SIG_IGUAL
              ;

lista_valores ::= valores end_lista_valores
              ;

end_lista_valores ::= COMA lista_valores

```

```
        | /* Nada */  
        ;  
  
valores ::= CADENA  
        | numero  
        ;  
  
numero  ::= ENTERO  
        | DECIMAL  
        ;
```

### **Diagrama de clases:**

Al nivel de este documento en la carpeta "documentación" se incluye el diagrama de clases.