

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2
MSC. LUIS ESPINO
AUX. JUAN CARLOS MAEDA
GRUPO 18**



TytusDB:Manual Usuario

ALEX RENÉ LÓPEZ ROSA	201602999
MIAMIN ELIEL BARRIOS ARRIVILLAGA	201603016
KEVIN GOLWER ENRIQUE RUIZ BARBALES	201603009
EDWARD DANILO GÓMEZ HERNÁNDEZ	201602909

Índice

Índice	2
Objetivos	3
Alcance	3
Introducción	3
PARTES Y FLUJO DE LA APLICACIÓN	4
Aplicación en funcionamiento	8
Conclusión	10

Objetivos

- **General:** Se espera ser una guía para cualquier usuario que desee hacer uso de la aplicación, indicando la manera correcta de utilizarla.
- **Específico:** Mostrar de manera gráfica cada una de las funcionalidades de la aplicación como también algunos ejemplos del funcionamiento.

Alcance

Se espera que esta guía pueda ayudar a cualquier tipo de usuario que planea hacer uso de la aplicación.

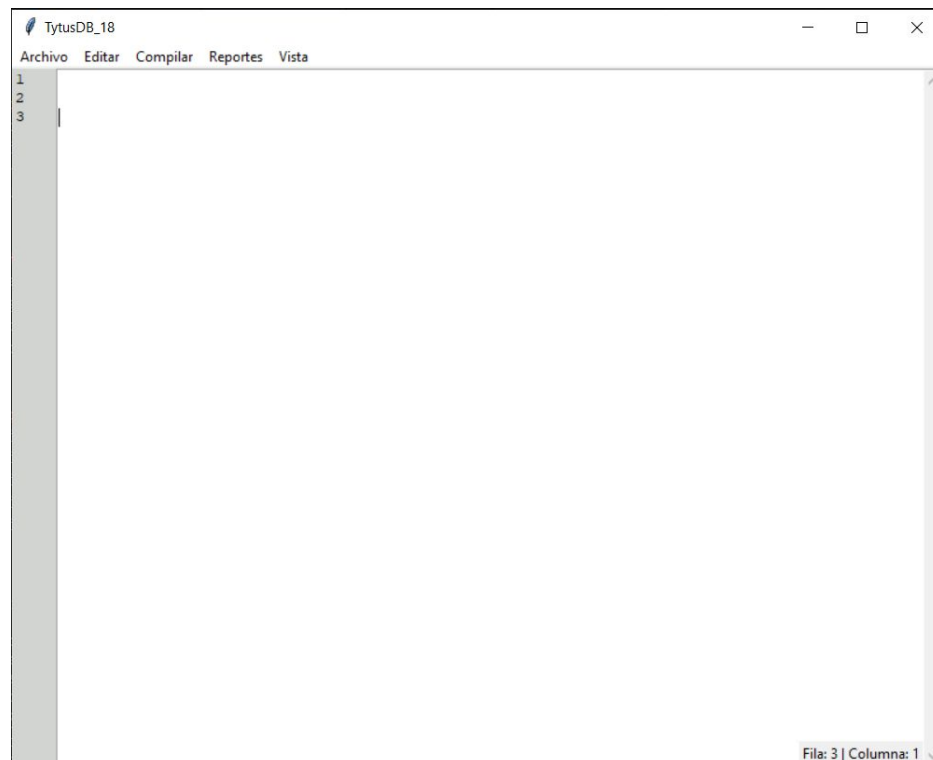
Introducción

Se espera que por medio de este documento el usuario pueda entender de mejor manera el funcionamiento de la aplicación, para tener una óptima experiencia al utilizarla.

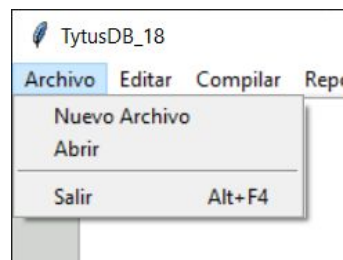
PARTES Y FLUJO DE LA APLICACIÓN

La aplicación se divide en los siguientes módulos:

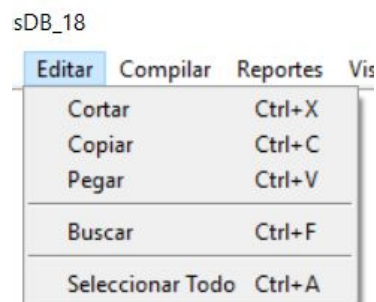
1. **Editor de texto:** El área de trabajo cuenta con un cuadro de texto en el cual se puede visualizar el número de fila y columna en la cual nos encontramos. En esta área se escriben todas las queries que se deseen



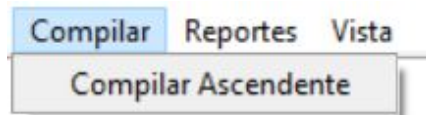
2. **Menù Archivo:** Este menú cuenta con las siguientes funcionalidades
 - **Nuevo Archivo:** Nos permite crear un archivo en blanco para poder escribir las queries que se necesiten
 - **Abrir:** Despliega un gestor de archivos para poder hacer una carga de un tipo de archivo .sql para poder ser leído su contenido y ser insertado en el área de trabajo
 - **Salir:** Nos permite salir de la aplicación



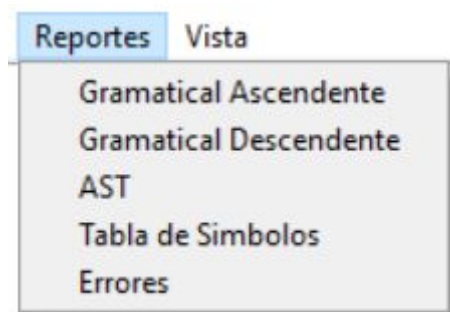
3. Menú Editar: Contiene funcionalidades básicas



4. Menú compilar: Esta opción nos permite analizar la entrada de las queries solicitadas en el área de trabajo.



5. Menú Reportes: Los reportes disponibles son los siguientes:



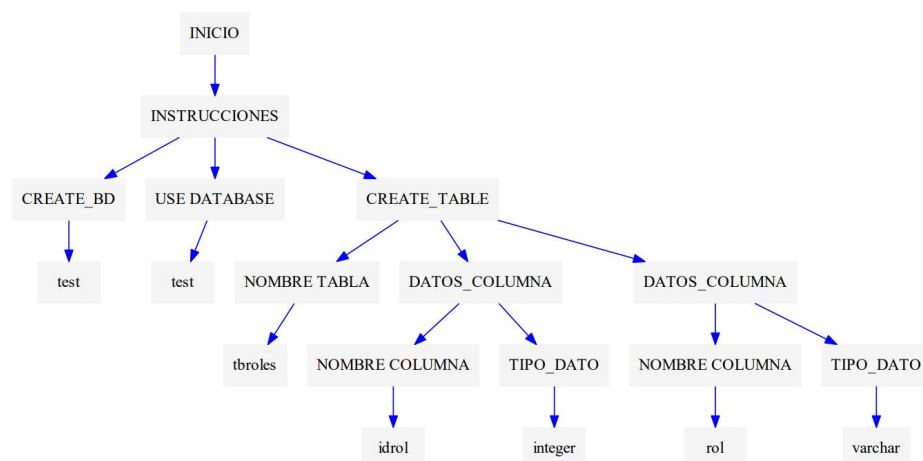
- **Reportes de errores léxico, sintácticos y semánticos.** Muestra el tipo, la descripción y el número de línea de los diferentes errores que se encontraron al momento de la ejecución

#	Listado de Errores
1	Error Semantico: 42P01: No existe la tabla para la herencia:citiess

- **Reporte de tabla de símbolos.** Muestra las variables, funciones y procedimientos con mínimo los siguientes datos: identificador, tipo, dimensión, declarada en, y referencias.

REPORTE TABLA DE SIMBOLOS				
instruccion	identificador	tipo	referencia	dimension
INSERT	val1	integer	llaves1	1
INSERT	val1	integer	llaves1	1
INSERT	val2	varchar	llaves2	1
	val3	integer	llaves2	1
INSERT	val2	varchar	llaves2	1
	val3	integer	llaves2	1
INSERT	val3	integer	mitabla	1
	val2	varchar	mitabla	1
	val1	integer	mitabla	1
INSERT	val1	integer	mitabla	1
	val2	varchar	mitabla	1
	val3	integer	mitabla	1

- **Reporte de AST.** Muestra el árbol de sintaxis abstracta utilizando Graphviz en una nueva ventana.



- **Reporte gramatical.** Abre archivo con Markdown que muestra las dos gramáticas con sintaxis BNF. En otro documento se muestra la

definición dirigida por la sintaxis con la gramática ascendente, indicando que expresiones se utilizaron, precedencia, símbolos terminales y no terminales, y las reglas semánticas.

```

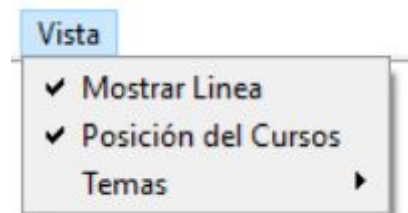
Preview gramatica_ASC.md X
Reporte Gramatical Ascendente

<init> ::= <l_sentencias> { init.val = l_sentencias.val }
<l_sentencias> ::= <l_sentencias sentencias> { l_sentencias.val = l_sentencias.append(sentencias.val) }
| <sentencias>
<sentencias> ::= <sentencia> ";" { sentencias.val = sentencias.val }
<sentencia> ::= <sentencia_ddl> { sentencia.val = sentencia_ddl.val }
| <sentencia_dml> { sentencia = sentencia_dml.val }
<sentencia_ddl> ::= <crear> { sentencia_ddl.val = crear.val }
| <liberar> { sentencia_ddl.val = liberar.val }
<sentencia_dml> ::= <insertar> { sentencia_dml.val = insertar.val }
| <actualizar> { sentencia_dml.val = actualizar.val }
| <eliminar> { sentencia_dml.val = eliminar.val }
| <seleccionH> { sentencia_dml.val = seleccionH.val }
| <mostrar> { sentencia_dml.val = mostrar.val }
| <altert> { sentencia_dml.val = altert.val }
| <usar> { sentencia_dml.val = usar.val }
<crear> ::= "CREATE" <reemplazar> "DATABASE" <verificacion> <ID> <propietario> <modo> { crear.val =
CrearBD(t[2], t[4], Operando_ID( test ), t[6], t[7]) }
<usar> ::= "USE" <ID> { usar.val = DBElegida(Operando_ID( test ))}
<crear> ::= "CREATE" "TABLE" <ID> "(" <columnas> ")" <herencia> { crear.val = CrearTabla(Operando_ID( troles
),t[7],t[5]) }

```

6. **Menú Vista:** El menú vista nos proporciona las siguientes funcionalidades:

- **Mostrar línea**
- **Posición del cursor**
- **Temas:** No permite escoger entre dos temas: Light o dark



Aplicación en funcionamiento

- **Create**

```
CREATE DATABASE IF NOT EXISTS test
  OWNER = 'root'
  MODE = 1;

USE test;

CREATE TABLE tbusuario (
  idusuario integer NOT NULL primary key,
  nombre varchar(50),
  apellido varchar(50),
  usuario varchar(15) UNIQUE NOT NULL,
  password varchar(15) NOT NULL,
  fechacreacion date
);

CREATE TYPE tipodato AS ENUM('Entero','Cadena','Boolean');
```

Consola

```
> Creando base de datos: test
  Todo OK
> Seleccionando base de datos: test
  Base de datos seleccionada
> Creando Tabla:tbusuario
  Todo OK
> Creacion de Type: tipodato
  Type registrado con exito
  con valores: ['Entero', 'Cadena', 'Boolean']
```

- **Insert**

```
idusuario integer NOT NULL primary key,
  nombre varchar(50),
  apellido varchar(50),
  usuario varchar(15) UNIQUE NOT NULL,
  password varchar(15) NOT NULL,
  fechacreacion date
);

CREATE TABLE tbroles (
  idrol integer NOT NULL primary key,
  rol varchar(15)
);
DROP TABLE tbroles;

CREATE TABLE tbrol (
  idrol integer NOT NULL primary key,
  rol varchar(15)
);

CREATE TABLE tbrolxusuario (
  idrol integer NOT NULL ,
  idusuario integer NOT NULL
);

insert into tbrol values (1,'Administrador');
insert into tbrol values (2,'Admin');
insert into tbrol values (3,'Ventas');
```

Consola

```
> Creando base de datos: test
  Todo OK
> Seleccionando base de datos: test
  Base de datos seleccionada
> Creando Tabla:tbusuario
  Todo OK
> Creando Tabla:tbroles
  Todo OK
> Eliminar Tabla:tbroles
  Tabla eliminada
> Creando Tabla:tbrol
  Todo OK
> Creando Tabla:tbrolxusuario
  Todo OK
> Insertado en Tabla:tbrol
  valores insertados:[1, 'Administrador']
> Insertado en Tabla:tbrol
  valores insertados:[2, 'Admin']
> Insertado en Tabla:tbrol
  valores insertados:[3, 'Ventas']
```


• Update y Delete

<pre> CREATE DATABASE IF NOT EXISTS test OWNER = 'root' MODE = 1; USE test; CREATE TYPE area AS ENUM ('CONTABILIDAD', 'ADMINISTRACION', 'VENTAS'); CREATE TABLE tbempleadopuesto (idempleado integer not null primary key, idpuesto integer not null, departamento varchar(50)); insert into tbempleadopuesto values(1,1,'ADMINISTRACION'); insert into tbempleadopuesto values(2,1,'CONTABILIDAD'); insert into tbempleadopuesto values(3,3,'CONTABILIDAD'); insert into tbempleadopuesto values(4,6,'VENTAS'); insert into tbempleadopuesto values(5,6,'VENTAS'); select * from tbempleadopuesto; UPDATE tbempleadopuesto SET idpuesto = 2 where idempleado = 2; select * from tbempleadopuesto; DELETE from tbempleadopuesto where idempleado=2; select * from tbempleadopuesto; </pre>	<pre> valores insertados:[5, 6, 'VENTAS'] > Select +-----+-----+-----+ idempleado idpuesto departamento +-----+-----+-----+ 1 1 ADMINISTRACION 2 1 CONTABILIDAD 3 3 CONTABILIDAD 4 6 VENTAS 5 6 VENTAS +-----+-----+-----+ > Actualizar tabla tbempleadopuesto Registro actualizado. > Select +-----+-----+-----+ idempleado idpuesto departamento +-----+-----+-----+ 1 1 ADMINISTRACION 2 2 CONTABILIDAD 3 3 CONTABILIDAD 4 6 VENTAS 5 6 VENTAS +-----+-----+-----+ > Eliminar tabla tbempleadopuesto Registro eliminado. > Select +-----+-----+-----+ idempleado idpuesto departamento +-----+-----+-----+ 1 1 ADMINISTRACION 3 3 CONTABILIDAD 4 6 VENTAS 5 6 VENTAS +-----+-----+-----+ </pre>
---	---

• Select

<pre> CREATE DATABASE IF NOT EXISTS test OWNER = 'root' MODE = 1; USE test; CREATE TYPE area AS ENUM ('CONTABILIDAD', 'ADMINISTRACION', 'VENTAS'); CREATE TABLE tbempleadopuesto (idempleado integer not null, idpuesto integer not null, departamento varchar(50)); insert into tbempleadopuesto values(1,1,'ADMINISTRACION'); insert into tbempleadopuesto values(2,1,'CONTABILIDAD'); insert into tbempleadopuesto values(3,3,'CONTABILIDAD'); insert into tbempleadopuesto values(4,6,'VENTAS'); insert into tbempleadopuesto values(5,6,'VENTAS'); select * from tbempleadopuesto; select idempleado, departamento from tbempleadopuesto; select abs(-15) as 'absoluto', cbirt(36) as 'Raiz'; </pre>	<pre> valores insertados:[3, 3, 'CONTABILIDAD'] > Insertado en Tabla:tbempleadopuesto valores insertados:[4, 6, 'VENTAS'] > Insertado en Tabla:tbempleadopuesto valores insertados:[5, 6, 'VENTAS'] > Select +-----+-----+-----+ idempleado idpuesto departamento +-----+-----+-----+ 1 1 ADMINISTRACION 2 1 CONTABILIDAD 3 3 CONTABILIDAD 4 6 VENTAS 5 6 VENTAS +-----+-----+-----+ > Select +-----+-----+ idempleado departamento +-----+-----+ 1 ADMINISTRACION 2 CONTABILIDAD 3 CONTABILIDAD 4 VENTAS 5 VENTAS +-----+-----+ > Select +-----+ absoluto +-----+ 15 +-----+ +-----+ Raiz +-----+ 3.3019272488946263 +-----+ </pre>
--	--

CABECERA Raiz RESULTADO 3.3019272488946
 <expresiones Operacion Math Unaria obje

Conclusión

Se espera que con la explicación realizada durante este documento el usuario haya comprendido de una mejor manera el funcionamiento de la aplicación tanto en su interfaz como también en la manera en que puede realizar el correcto funcionamiento.