

Manual de Usuario - Analizador Léxico y Sintáctico de SQL

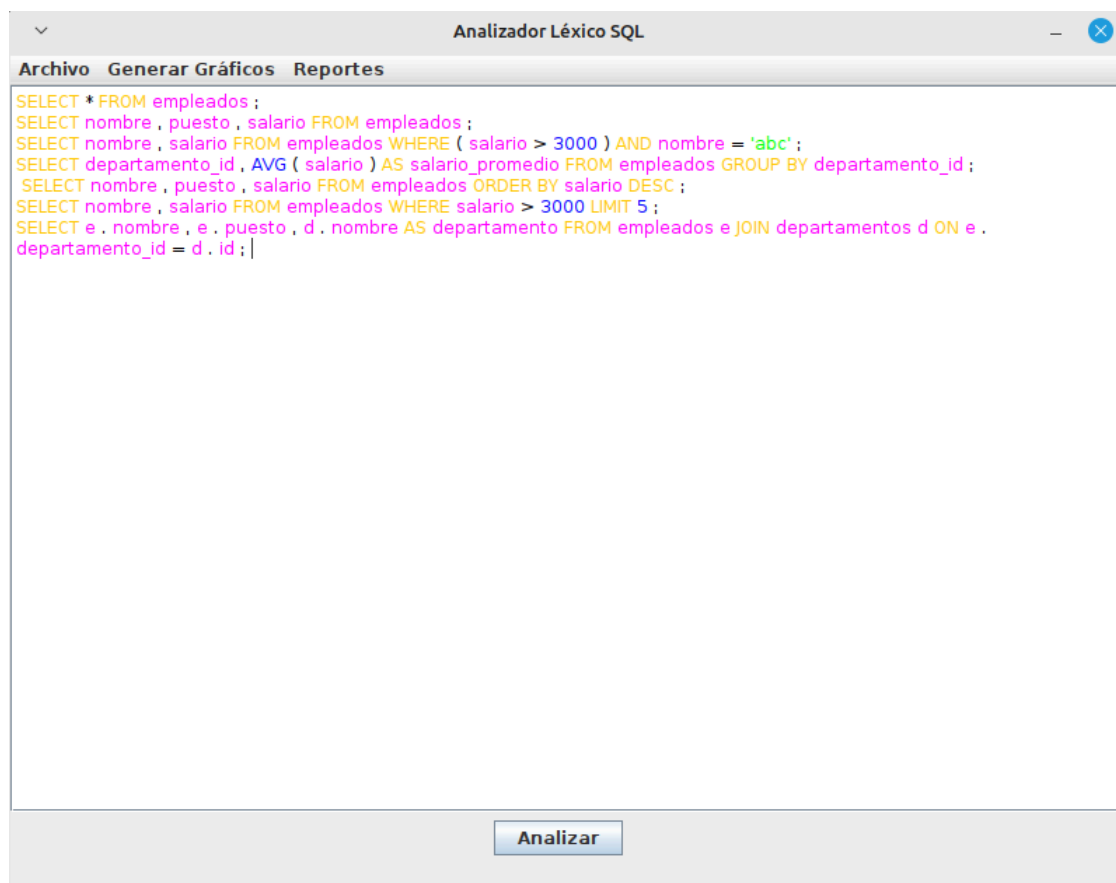
1. Introducción

El analizador léxico y sintáctico está diseñado para aceptar estructuras DDL (Data Definition Language) y DML (Data Manipulation Language) del lenguaje SQL. El programa incluye un editor de código que resalta los tokens según su tipo, facilitando la visualización y comprensión del código SQL.

2. Interfaz de Usuario

La interfaz incluye:

- **Editor de Código:** Permite escribir y editar código SQL.
- **Coloreado de Tokens:** Los tokens se colorean según su tipo, como se especifica a continuación.
- **Opciones de Archivo:** Permite cargar, crear, guardar, y guardar como archivos de texto.



2.1. Colores de Tokens

Tipo de Token	Color	Observación
Palabras Reservadas	Naranja	Incluye: CREATE, DATABASE, TABLE, etc.
Tipo de Dato	Morado	Incluye: INTEGER, VARCHAR, DATE, etc.
Números Enteros	Azul	Ejemplos: 100, 50, 2, 3, 25
Números Decimales	Azul	Ejemplos: 65.50, 52.02, 6000.558
Fechas	Amarillo	Formato: 'YYYY-MM-DD', debe estar entre comillas.
Cadenas	Verde	Deben estar entre comillas simples.
Identificadores	Fucsia	Uso de snake case, solo letras minúsculas y guiones bajos.
Booleanos	Azul	Valores: TRUE, FALSE
Funciones de Agregación	Azul	Ejemplos: SUM, AVG, COUNT, etc.
Signos y Operadores	Negro	Incluye: (,), , , ; , =, +, -, *, /, <, >, AND, OR, NOT
Comentarios de Línea	Gris	Comienzan con '--'

3. Sintaxis Soportada

3.1. Estructuras DDL

Creación de Base de Datos

```
CREATE DATABASE <identificador>;
```

Ejemplo:

```
CREATE DATABASE store;
```

Creación de Tablas

```
CREATE TABLE <identificador> (  
    <estructura_de_declaracion>,
```

```
    [estructura_de_llaves]
);
```

Ejemplo:

```
CREATE TABLE empleados (
    id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    salario DECIMAL(10, 2),
    fecha_contratacion DATE
);
```

Modificaciones de Tablas

```
ALTER TABLE <identificador> ADD COLUMN <identificador>
<tipo_de_dato>;
ALTER TABLE <identificador> DROP COLUMN <identificador>;
```

Ejemplo:

```
ALTER TABLE empleados ADD COLUMN fecha_nacimiento DATE;
```

Eliminación de Tablas

```
DROP TABLE IF EXISTS <identificador> CASCADE;
```

3.2. Estructuras DML

Inserción de Datos

```
INSERT INTO <identificador> (<identificador>, ...) VALUES ([DATO],
...);
```

Ejemplo:

```
INSERT INTO empleados (nombre, salario) VALUES ('Juan Pérez',
3500.00);
```

Lectura de Datos

```
SELECT * FROM <identificador> [SENTENCIAS];
```

Ejemplo:

```
SELECT nombre, salario FROM empleados;
```

Actualización de Datos

```
UPDATE <identificador> SET <identificador> = [DATO] [WHERE];
```

Ejemplo:

```
UPDATE empleados SET salario = 4000.00 WHERE id = 1;
```

Eliminación de Datos

```
DELETE FROM <identificador> [WHERE];
```

Ejemplo:

```
DELETE FROM empleados WHERE id = 3;
```

4. Reporte de Errores

Los errores léxicos y sintácticos se reportan en tablas que incluyen:

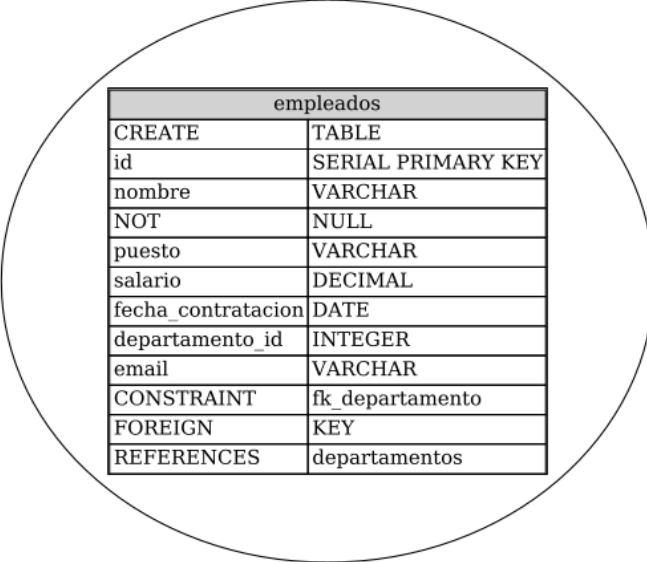
- **Errores Léxicos:** Token, Línea, Columna, Descripción.
- **Errores Sintácticos:** Token, Tipo de Token, Línea, Columna, Descripción.

Ejemplo de reporte:

Token	Línea	Columna	Descripción
<>	1	3	Token no reconocido
TBL	7	8	Token no reconocido

5. Generación de Gráficos

La generación de gráficos crea imágenes que representan las estructuras sql en un archivo html que se genera en la ruta relativa del programa.



empleados	
CREATE	TABLE
id	SERIAL PRIMARY KEY
nombre	VARCHAR
NOT	NULL
puesto	VARCHAR
salario	DECIMAL
fecha_contratacion	DATE
departamento_id	INTEGER
email	VARCHAR
CONSTRAINT	fk_departamento
FOREIGN	KEY
REFERENCES	departamentos