



UNIVERSIDAD AUTÓNOMA DE TAMAULIPAS

FACULTAD DE INGENIERÍA TAMPICO

INTEGRANTES:

**FERNANDEZ ARRAZOLA SOFIA MARIANA
GONZÁLEZ GABRIEL URIEL
MARTÍNEZ OROZCO ALAN ENRIQUE
CISNEROS GARCÍA NATALIA**

DOCENTE:

MUÑOZ QUINTERO DANTE ADOLFO

ASIGNATURA:

PROGRAMACIÓN DE SISTEMAS DE BASE 2

GRADO Y GRUPO:

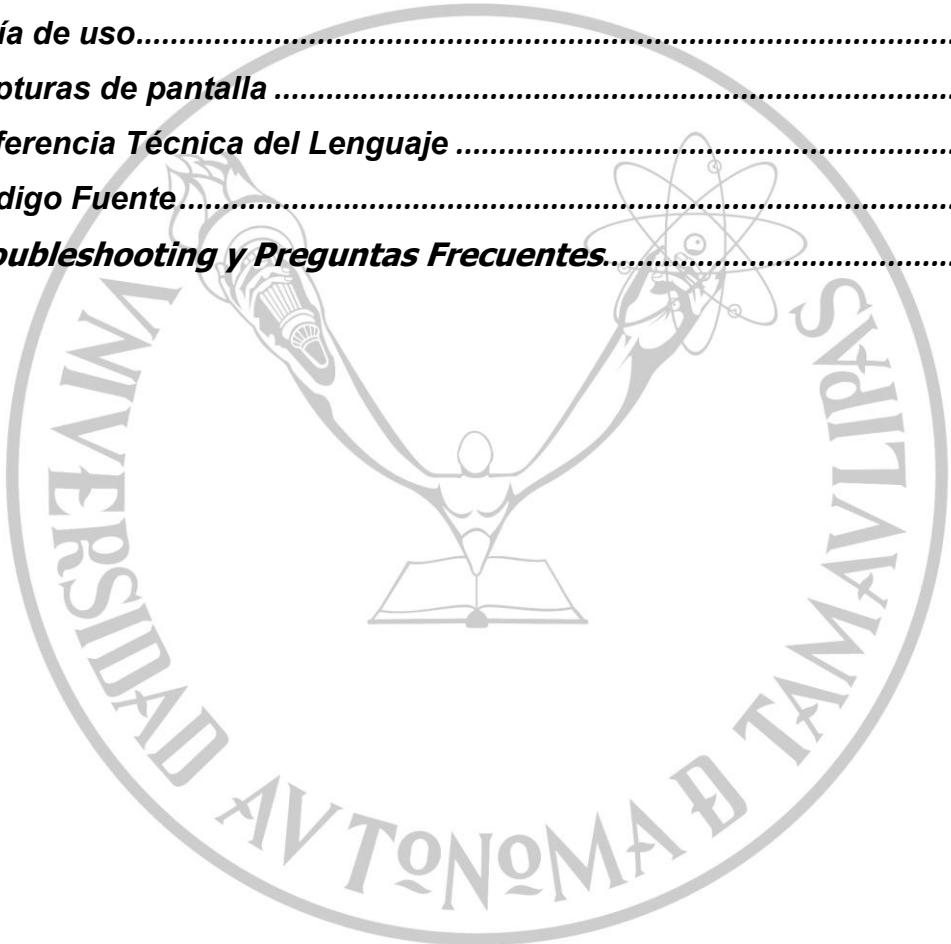
9º “J”

VERDAD, BELLEZA, PROBIDAD

PRODUCTO INTEGRADOR

ÍNDICE

1. <i>Introducción</i>	3
2. <i>Instalación y configuración</i>	4
3. <i>Guía de uso</i>	5
4. <i>Capturas de pantalla</i>	9
5. <i>Referencia Técnica del Lenguaje</i>	12
6. <i>Código Fuente</i>	13
7. <i>Troubleshooting y Preguntas Frecuentes</i>	15



VERDAD, BELLEZA, PROBIDAD

1. Introducción

Este proyecto consiste en un Traductor de Consultas SQL Simples elaborados con lenguaje Python, una aplicación diseñada para procesar instrucciones en un lenguaje de natural y convertirlas a sentencias SQL estándar compatibles con bases de datos relacionales.

El objetivo principal de este traductor es ofrecer una interfaz más intuitiva y limitada para que usuarios no familiarizados con la sintaxis de SQL puedan realizar operaciones básicas de consulta y manipulación de datos. En lugar de escribir sentencias complejas como SELECT * FROM, el usuario interactúa con comandos más sencillos y directos, como MOSTRAR, CONTAR o BORRAR

VERDAD, BELLEZA, PROBIDAD

2. Instalación y configuración

Requisitos Previos

- Python: Se requiere la versión Python 3.x.

Verificación: Ejecutar `python --version` o `python3 --version` en la terminal.

- Java Runtime Environment (JRE): Necesario para ejecutar la herramienta ANTLR (aunque solo se usa para generar el código, puede ser relevante si el usuario quiere modificar la gramática).

Verificación: Ejecutar `java -version`.

Instalación de dependencias

Instalar ANTLR4 Runtime para Python:

```
pip install antlr4-python3-runtime
```

Configuración del Entorno de Desarrollo

A. Descargar del ANTLR Tool JAR

- Descarga el archivo `antlr-VERSION-complete.jar` desde el sitio oficial de ANTLR.
- Guardar el archivo en una ubicación accesible.

B. Generación de Clases del Parser y Lexer

Una vez que el archivo `.jar` esté en el proyecto, se puede generar las clases necesarias de Python a partir de la gramática.

- Abrir la terminal en la carpeta donde se encuentra la gramática.
- Ejecutar el siguiente comando, reemplazando `NOMBRE_ARCHIVO.jar` por el nombre del archivo que descargaste (ej: `antlr-4.13.1-complete.jar`):

```
java -jar NOMBRE_ARCHIVO.jar -Dlanguage=Python3 ConsultaSQL.g4 -visitor
```

Verificar que los siguientes archivos hayan sido generados en la carpeta:

- ConsultaSQLLexer.py
 - ConsultaSQLParser.py
 - ConsultaSQLVisitor.py

Ejecución del Proyecto

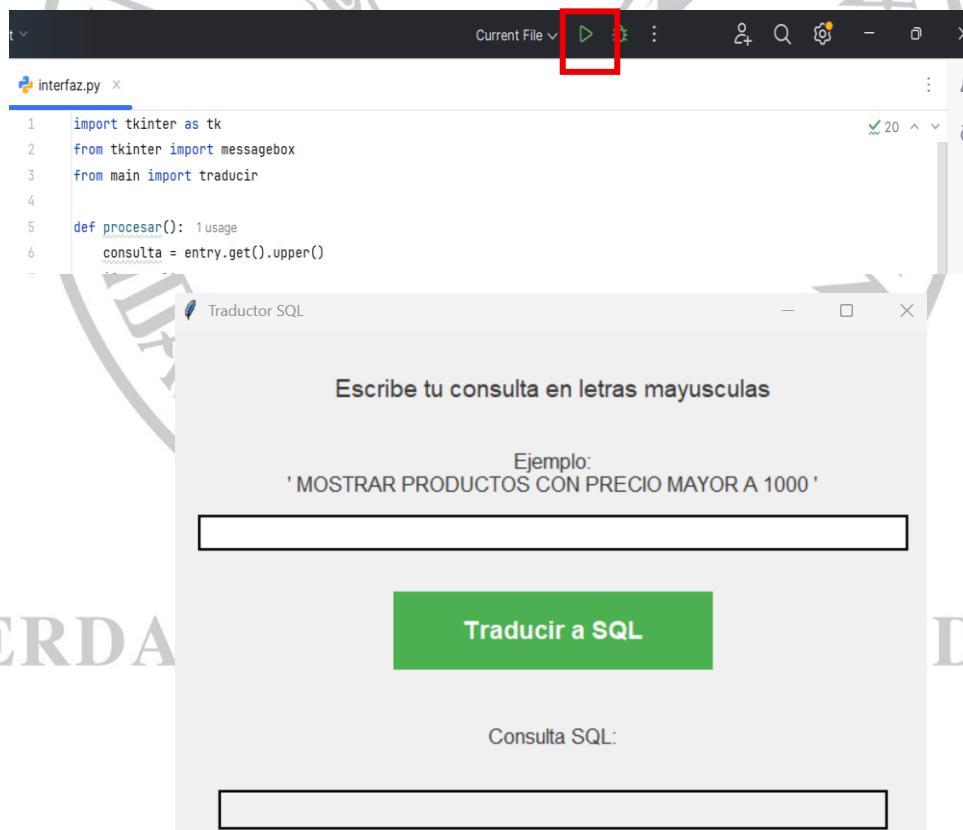
A. Ejecución de la Interfaz Gráfica

Para usar la aplicación visualmente, se ejecuta el archivo de la interfaz.

3. Guía de uso

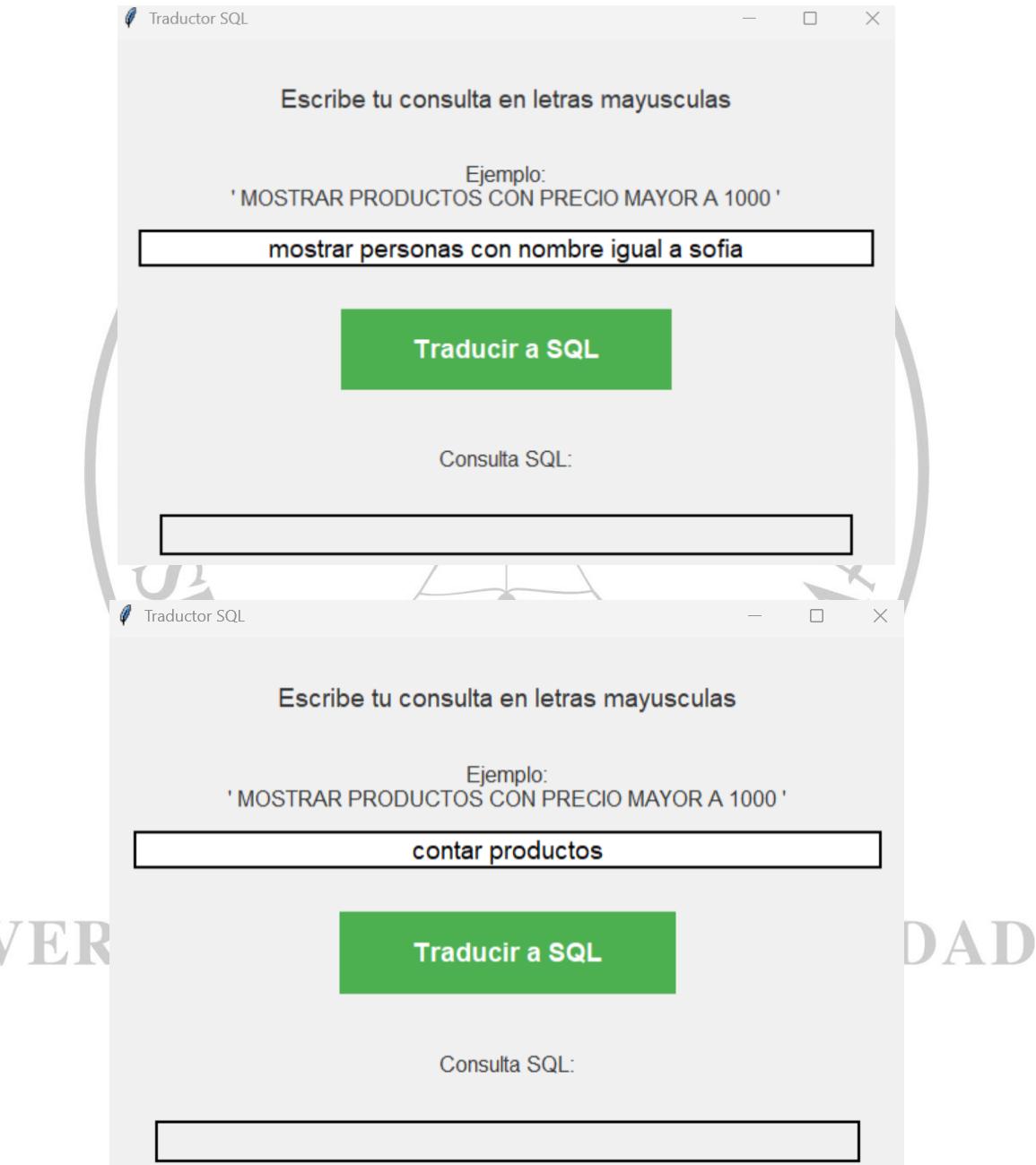
Guía de uso

1. Iniciar la Aplicación



2. Escribe tu comando en el campo de texto:

Nota: Todas las palabras clave (MOSTRAR, CONTAR, BORRAR, CON, DONDE, SEA, MAYOR A, MENOR A, IGUAL A) y los identificadores (nombres de tablas y campos) deben escribirse siempre en mayúsculas.



Escribe tu consulta en letras mayusculas

Ejemplo:
' MOSTRAR PRODUCTOS CON PRECIO MAYOR A 1000 '

mostrar personas con nombre igual a sofia

Traducir a SQL

Consulta SQL:

Escribe tu consulta en letras mayusculas

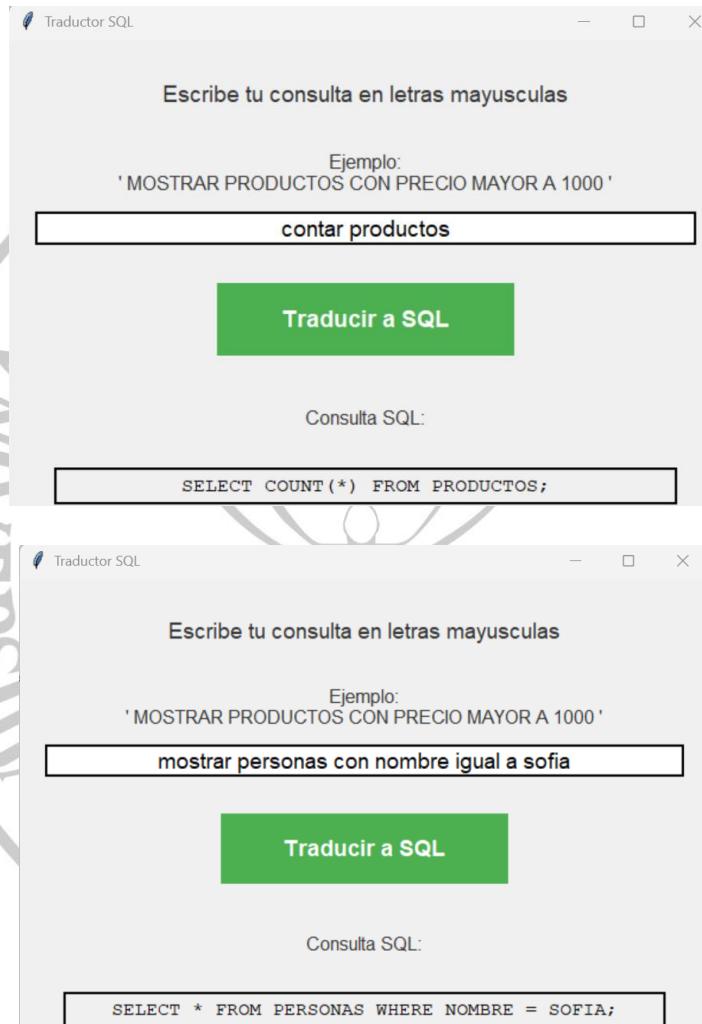
Ejemplo:
' MOSTRAR PRODUCTOS CON PRECIO MAYOR A 1000 '

contar productos

Traducir a SQL

Consulta SQL:

- 3.** Haz clic en el botón "Traducir a SQL"
- 4.** La sentencia SQL estándar resultante aparecerá en el cuadro de texto inferior



The image shows two instances of the "Traductor SQL" application window. Both windows have a light gray background with a watermark of the University of Tamaulipas logo (a circular seal with the text "UNIVERSIDAD DE TAMAULIPAS" around the perimeter and a central figure) overlaid on them.

Top Window (Example 1):

- Text input field: "contar productos"
- Green button: "Traducir a SQL"
- Output field: "Consulta SQL:
SELECT COUNT(*) FROM PRODUCTOS;"

Bottom Window (Example 2):

- Text input field: "mostrar personas con nombre igual a sofia"
- Green button: "Traducir a SQL"
- Output field: "Consulta SQL:
SELECT * FROM PERSONAS WHERE NOMBRE = SOFIA;"

VERDAD, BELLEZA, PROBIDAD

a) Mostrar

Se utiliza para recuperar todos los datos de una tabla. Puede ser simple o incluir una condición de filtrado.

MOSTRAR [NOMBRE_TABLA]

Ejemplo	Traducción
<i>MOSTRAR PRODUCTOS</i>	<i>SELECT * FROM PRODUCTOS;</i>

Con WHERE:

*MOSTRAR [NOMBRE_TABLA] CON [NOMBRE_CAMPO] [COMPARADOR]
[VALOR]*

Ejemplo	Traducción
<i>MOSTRAR EMPLEADOS CON EDAD MAYOR A 30</i>	<i>SELECT * FROM EMPLEADOS WHERE EDAD > 30;</i>

b) Contar

CONTAR [NOMBRE_TABLA]

Ejemplo	Traducción
<i>CONTAR PRODUCTOS</i>	<i>SELECT COUNT (*) FROM CLIENTES;</i>

c) Borrar

*BORRAR [NOMBRE_TABLA] DONDE [NOMBRE_CAMPO] SEA
[NUMERO]*

VERDAD, BELLEZA, PROBIDAD

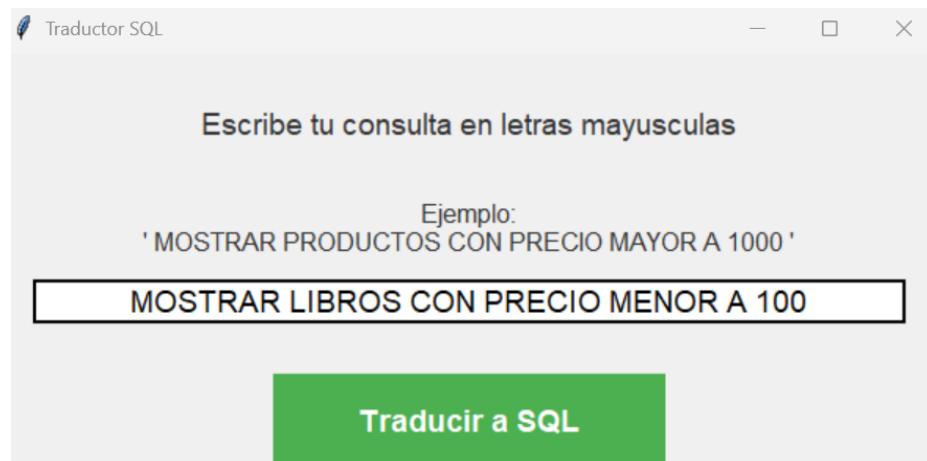
d) Manejo de errores

Causa del Error	Mensaje Típico	Solución
Usar minúsculas en palabras clave.	Error: ...token recognition error...	Asegúrate de que todas las palabras clave estén en MAYÚSCULAS.
Faltar un campo o valor en una condición.	Error: ...missing 'ID' at 'DONDE'...	Revisa que todos los componentes de la regla se hayan incluido (Ej: MOSTRAR TABLA CON CAMPO IGUAL A VALOR).
Usar un identificador con números.	Error: ...token recognition error...	Identificadores (ID) solo deben contener letras mayúsculas ([A-Z]+).

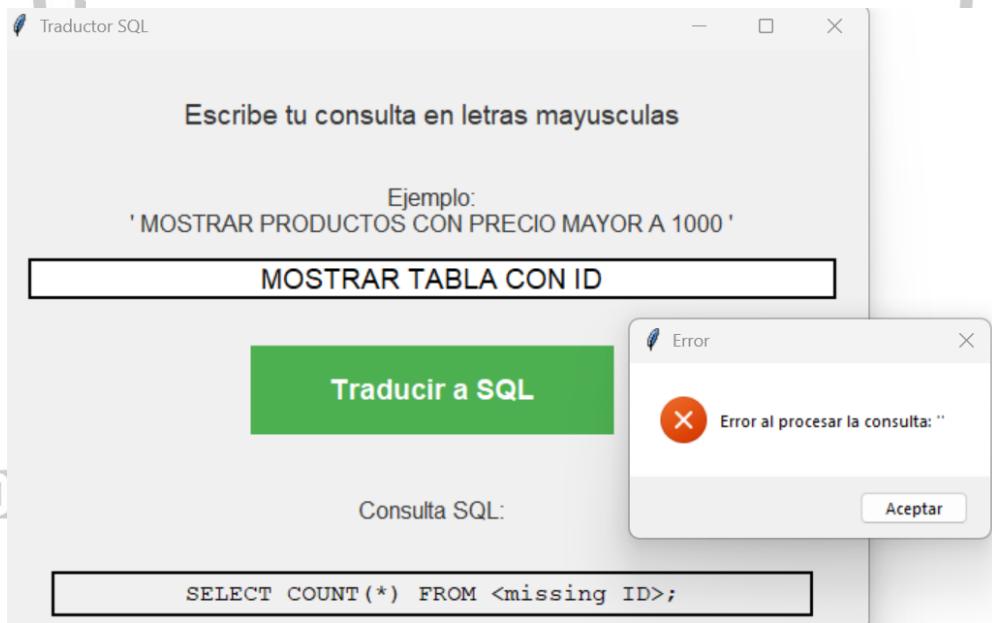
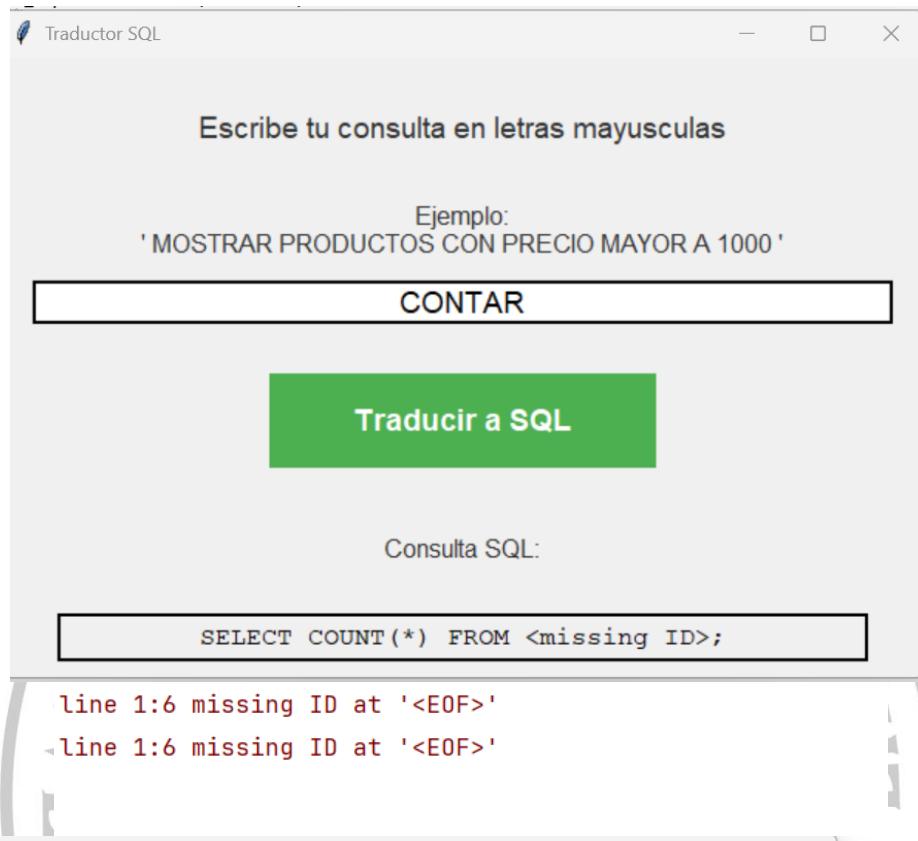
4. Capturas de pantalla

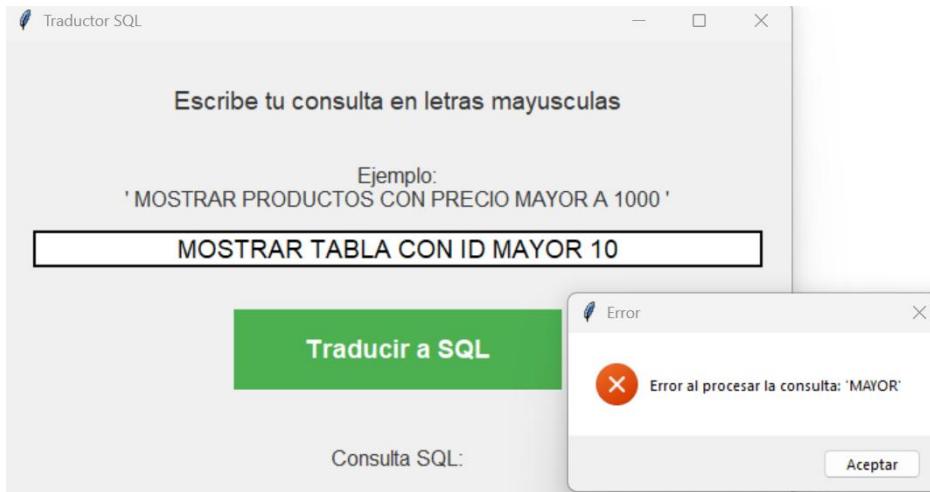


The screenshot shows a window titled "Traductor SQL". Inside, there's a text input field with the placeholder "Escribe tu consulta en letras mayusculas" containing the query "Borrar Pedidos donde ID_PEDIDO sea 45". Below this, a green button says "Traducir a SQL". At the bottom, the translated SQL query is displayed: "DELETE FROM PEDIDOS WHERE ID = 45;".



VEI DAD





```
line 1:21 mismatched input 'MAYOR' expecting {'MAYOR A', 'MENOR A', 'IGUAL A'}
line 1:21 mismatched input 'MAYOR' expecting {'MAYOR A', 'MENOR A', 'IGUAL A'}
```

5. Referencia Técnica del Lenguaje

Componentes léxicos

Token	Descripción	Regla ANTLR	Ejemplo
Palabras Clave	Palabras reservadas que definen la operación. Deben ir en mayúsculas.	Tokens literales ('MOSTRAR', 'CONTAR', etc.)	MOSTRAR, DONDE, IGUAL A
ID	Identificador para nombres de tablas o campos. Solo letras mayúsculas.	[A-Z]+	PRODUCTOS, NOMBRE
NUM	Números enteros positivos para valores de condición.	[0-9]+	100, 45, 3
WS	Espacios en blanco, tabulaciones y saltos de línea.	[\t\r\n]+ -> skip	(Ignorado)

Reglas semánticas

Regla	Expansión	Descripción
consulta	mostrar contar borrar	La consulta debe ser exactamente una de estas tres operaciones.
mostrar	'MOSTRAR' ID (condición)?	Sentencia para seleccionar datos. El campo condición es opcional.
contar	'CONTAR' ID	Sentencia para contar registros. Requiere solo el nombre de la tabla.
borrar	'Borrar' ID 'DONDE' ID 'SEA' NUM	Sentencia de eliminación. Requiere una condición estricta de igualdad numérica.
condición	'CON' ID comparador valor	La cláusula WHERE del MOSTRAR.
comparador	'MAYOR A' 'MENOR A' 'IGUAL A'	Las tres opciones de operadores relacionales.
valor	NUM	El valor que se compara (solo acepta números).

6. Código Fuente

```
ConsultaSQL.g4 x
1 grammar ConsultaSQL;
2 consulta
3   : mostrar
4   | contar
5   | borrar
6   ;
7 mostrar
8   : 'MOSTRAR' ID (condicion)?
9   ;
10 contar
11  : 'CONTAR' ID
12  ;
13 borrar
14  : 'Borrar' ID 'DONDE' ID 'SEA' NUM
15  ;
16 condicion
17  : 'CON' ID comparador valor
18  ;
19 comparador
20  : 'MAYOR A' | 'MENOR A' | 'IGUAL A'
21  ;
22 valor
23  : NUM
24  ;
25 ID : [A-Z]+ ;
26 NUM : [0-9]+ ;
27 WS : [ \t\r\n]+ -> skip ;
```

```

main.py ×

1  from antlr4 import *
2  from ConsultaSQLLexer import ConsultaSQLLexer
3  from ConsultaSQLParser import ConsultaSQLParser
4  from ConsultaSQLVisitor import ConsultaSQLVisitor
5
6  class TraductorSQLVisitor(ConsultaSQLVisitor):  1 usage
7  @t  def visitMostrar(self, ctx):
8      tabla = ctx.ID().getText()
9
10     if ctx.condicion():
11         campo = ctx.condicion().ID().getText()
12         comp = ctx.condicion().comparador().getText()
13         valor = ctx.condicion().valor().getText()
14     >     operador = {...}[comp]
15         return f"SELECT * FROM {tabla} WHERE {campo} {operador} {valor};"
16     return f"SELECT * FROM {tabla};"
17
18 @t  def visitContar(self, ctx):
19     tabla = ctx.ID().getText()
20     return f"SELECT COUNT(*) FROM {tabla};"
21
22 @t  def visitBorrar(self, ctx):
23     tabla = ctx.ID(0).getText()
24     campo = ctx.ID(1).getText()
25     valor = ctx.NUM().getText()
26     return f"DELETE FROM {tabla} WHERE {campo} = {valor};"
27 def traducir(instrucion):  2 usages
28     input_stream = InputStream(instrucion)
29
30     stream = CommonTokenStream(lexer)
31
32     parser = ConsultaSQLParser(stream)
33
34     tree = parser.consulta()
35
36     visitor = TraductorSQLVisitor()
37
38     return visitor.visit(tree)
39
40
41
42
43
44

```

VERDAD, BELLEZA, PROBIDAD

7. Troubleshooting y Preguntas Frecuentes

¿Puedo usar mi traductor para hacer *JOINS* o *GROUP BY*?

No. El lenguaje está limitado a las tres operaciones definidas en la gramática (MOSTRAR, CONTAR, BORRAR). Para operaciones complejas, debe escribir el SQL directamente.

¿Por qué mis nombres de campos no pueden tener guiones bajos o números?

El token ID en la gramática se definió estrictamente como [A-Z]+ (solo letras mayúsculas). Si desea usar otros caracteres, la regla ID en ConsultaSQL.g4 debe modificarse.

¿El resultado de la traducción se conecta automáticamente a la base de datos?

No. El proyecto solo realiza la traducción de la sentencia. La sentencia SQL resultante debe ser tomada por el usuario y ejecutada en un cliente o programa de base de datos aparte.

VERDAD, BELLEZA, PROBIDAD