**Breve informe sobre el Entregable 3 al crear un DSL (Domain-Specific Language) llamado USQL**

**Componentes principales:**

Traductor de SQL a USQL y viceversa: Utilizando PLY, se implementó un parser (Es un componente que se encarga de analizar una secuencia de tokens para determinar su estructura conforme a una gramática formal) que traduce las consultas SQL al nuevo dialecto USQL, validando también que la consulta SQL generada sea correcta.

Fluent API (Es un estilo de programación orientado a hacer el código más legible y expresivo mediante el encadenamiento de métodos) en Python: se utilizó la gramática de PLY para definir las reglas del parser y se implementó una API fluida que permite crear consultas de manera clara, permitiendo encadenar filtros y operadores fácilmente.

**Inconvenientes Encontrados:**

Manejo de Tokens (Es la unidad más pequeña significativa para un analizador léxico o un parser. Ej: “Traeme”) No Definidos: Algunos tokens no estaban definidos explícitamente en la versión inicial de la lista de palabras clave de USQL. Esto generó algunos errores durante la traducción, y tuvimos que pensar una forma para mantener los tokens originales si no existía una traducción específica.

Ambigüedades en Traducciones: Algunas palabras clave no tenían una traducción directa clara, lo cual fue confuso y tuvimos que ajustarlo manualmente y probar para asegurarnos que los resultados estuvieran bien.

Cobertura de Pruebas: Usamos el paquete pycobertura (sirve para analizar y reportar la cobertura de pruebas del código Python) para evaluar la cobertura, y tuvimos que realizar una refactorización (reestructurar el código, sin cambiar su comportamiento externo) para incluir pruebas adicionales y asegurar la robustez del código.

Manejo de Errores: El manejo de excepciones nos complicó y tuvimos que realizar múltiples iteraciones para cubrir todos los posibles errores de sintaxis y casos de uso extremos.

**Módulos:**

* Usql\_translator.py (código principal)
* Parser.out (generado automáticamente por PLY y contiene información de depuración sobre la gramática y el parserrado)
* Parsetab.py (generado automáticamente por PLY durante la compilación del parser. Contiene estructuras de datos que representan la “gramática del lenguaje”, ayudando al parser comprender y procesar las consultas)
* Test.py (test unitarios para comprobar al menos el 85% de la cobertura)