

## **Informe técnico – Búsqueda Booleana en Documentos.**

El presente programa en Python implementa un sistema de búsqueda booleana sobre un pequeño conjunto de documentos escritos en español. Su objetivo principal es permitir al usuario realizar consultas lógicas utilizando operadores como AND, OR y NOT, para recuperar aquellos documentos que contienen o excluyen ciertas palabras clave.

### **Funcionamiento general:**

#### **1. Carga de documentos:**

Se define un diccionario con cinco documentos, cada uno representado como una cadena de texto.

#### **2. Preprocesamiento:**

Los textos se tokenizan, se convierten a minúsculas y se eliminan las palabras vacías (stopwords) del idioma español, usando recursos de la biblioteca NLTK. Este paso permite centrarse solo en los términos significativos.

#### **3. Índice invertido:**

Se construye un índice donde cada palabra relevante se asocia con los documentos en los que aparece. Este índice permite realizar búsquedas eficientes sin tener que escanear todos los textos cada vez.

#### **4. Consulta booleana:**

El usuario introduce una consulta utilizando operadores lógicos. El sistema interpreta los operadores (AND, OR, NOT) y evalúa los términos en función del índice invertido. La búsqueda se hace respetando el orden de aparición de los términos, pero no admite paréntesis ni jerarquía de operadores (precedencia lógica), lo cual podría ser una mejora futura.

#### **5. Interfaz interactiva:**

Mediante un bucle while, el programa solicita al usuario que ingrese consultas hasta que se escriba la palabra "salir". Los documentos que coinciden con la consulta se muestran como resultado.

**Observaciones:**

- El código es claro y está bien estructurado para un entorno educativo o demostrativo.
- Utiliza recursos básicos de NLTK, una biblioteca popular en procesamiento de lenguaje natural.
- No se contempla el manejo de errores o validación más profunda de entradas, aunque esto no afecta el funcionamiento en pruebas simples.

**Conclusión:**

Este trabajo representa una implementación funcional y didáctica de un buscador booleano. Refuerza conceptos como tokenización, filtrado de palabras irrelevantes, creación de índices y aplicación de lógica booleana sobre conjuntos. Su estructura modular y su interfaz interactiva lo hacen útil para experimentar con técnicas básicas de recuperación de información.