**Proyecto 2**

**Recuperación de Documentos de Texto**

1. **Introducción**

El logro del estudiante está enfocado a entender y aplicar los algoritmos de búsqueda y recuperación de la información basado en el contenido.

Este proyecto está enfocado a la **construcción óptima** de un Índice Invertido para tareas de búsqueda y recuperación en documentos de texto.

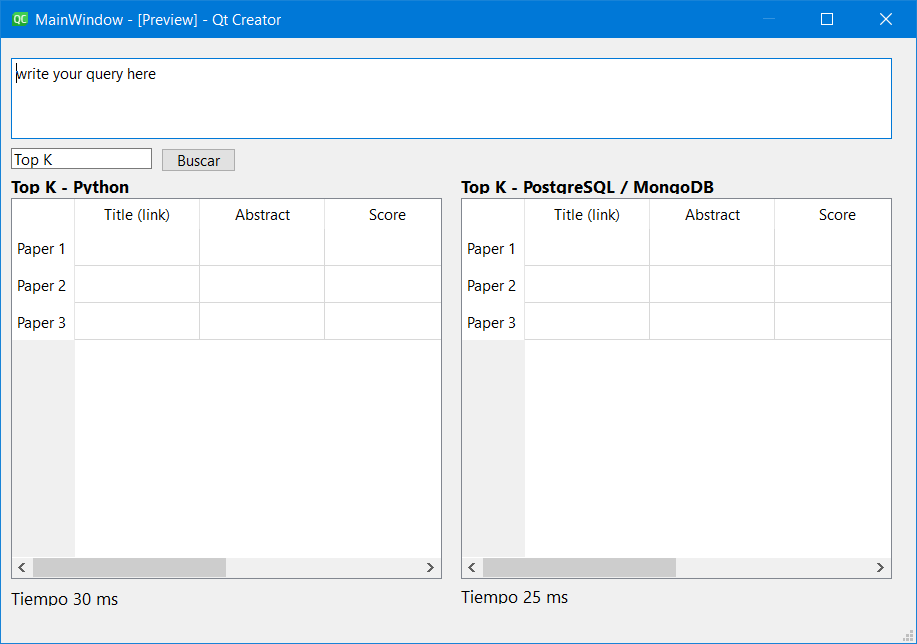


Figura 1: Aplicación para buscar papers.

1. **Backend**: Implementación del Índice Invertido

Implementar el índice invertido para recuperación de texto usando el modelo de recuperación por ranking para consultas de texto libre. Considere los siguientes pasos generales:

* Preprocesamiento:
  + Tokenization
  + Filtrar Stopwords
  + Reducción de palabras *(Stemming)*
* Construcción del Índice
  + Estructurar el índice invertido para guardar los pesos TF-IDF.
  + Calcular una sola vez la longitud de cada documento (norma) y guardarlo para reutilizarlo al momento de aplicar la similitud de coseno.
  + **Construcción del índice en memoria secundaria para grandes colecciones de datos.** 
    - Single-pass in-memory indexing (slide 56-59)
    - Si plantea otra versión del algoritmo para memoria secundaria, debe explicar con gráficos el funcionamiento.
* Consulta
  + La consulta es una frase en lenguaje natural.
  + El scoring se obtiene aplicando la similitud de coseno sobre el índice invertido en **memoria secundaria**.
  + La función de recuperación debe retornar una lista ordenada de documentos que se aproximen a la consulta.

1. **Frontend**: Full-Text Search

Para probar la funcionalidad del índice invertido, se debe construir una aplicación que permita interactuar con las principales operaciones del índice invertido:

* Opción para cargar e indexar todos los documentos.
  + También puede ir procesando los documentos en tiempo real (streaming).
* La búsqueda textual debe estar relacionado con el tema de interés.
* Indicar la cantidad de documentos a recuperar (Top K)
* Presentación de resultados de la búsqueda de forma amigable al usuario
  + Indicar el tiempo que toma la consulta
* Se debe **comparar con los resultados** que se obtiene de **PostgreSQL** (opcionalmente MongoDB)
  + Esto servirá para medir el performance de su implementación.
  + Investigue cómo PostgreSQL realiza la recuperación de textos: índice y función de similitud.

1. **Dataset**

El grupo tiene la libertad de escoger cualquier tópico de interés siempre y cuando se tenga un repositorio considerable de documentos. A continuación, se lista algunos repositorios de documentos en inglés.

* arXiv Dataset [recomendado]:

<https://www.kaggle.com/datasets/Cornell-University/arxiv>

* Twitter Dataset [recomendado]:

[twitter\_tracking](https://1drv.ms/f/s!Am-BH5_fIykMg4x0BGUVMoIK8O2c5A?e=6xI1Jj)

* All the News

<https://www.kaggle.com/snapcrack/all-the-news>

* Email Dataset

<https://www.kaggle.com/nitishabharathi/email-spam-dataset>

1. **Entregable**

* Los alumnos formaran grupos de máximo de cuatro integrantes.
* El proyecto estará alojado enteramente en GitHub.
* Trabajar de forma colaborativa, se considerará para su nota individual.
* En el Canvas subir solo el **enlace público** del proyecto.
* Incluir en el informe un cuadro de actividades por integrante.
* La fecha límite de entrega es el 15 de Junio.

1. **Informe del proyecto**

* Archivo Readme o Wiki en GitHub
* Ortografía y consistencia en los párrafos.
* El informe no debe ser extenso, pero debe describir todos aspectos importantes de la implementación.
  + [2 puntos] Descripción del dominio de datos
  + [12 puntos] Backend:
    - Construcción del índice invertido
    - Manejo de memoria secundaria
    - Ejecución óptima de consultas
  + [6 puntos] Frontend:
    - Diseño del índice con PostgreSQL/MongoDB
    - Análisis comparativo con su propia implementación
    - Screenshots de la GUI
* Debe incluir imágenes o diagramas para una mejor comprensión.
* ¡La **presentación del producto** se realizará en clase!