ESCUELA POLITECNICA NACIONAL

RECUPERACION DE LA INFORMACION

PROYECTO I BIM

STEVEN ERAZO, SEBASTIAN ROBLEZ, JORGE ROJAS

El proyecto será gestionado a través de **Git y GitHub**, siguiendo una estructura de ramas y tareas claras para asegurar un flujo de trabajo colaborativo, ordenado y eficiente. A continuación, se describe la organización acordada para el desarrollo.

Roles y Responsabilidades

- **Sebastián (Sebas)**: Responsable del **frontend** y dueño del repositorio principal. Es quien aprueba los *pull requests* y realiza el *merge* final en la rama principal.
- **Steven**: Encargado de la **limpieza y procesamiento del corpus de datos**, asegurando que el dataset esté listo para su análisis y uso.
- **Jorge**: Apoya en **todas las áreas** del proyecto, incluyendo integración del frontend, procesamiento del corpus y conexión entre módulos. Actúa como puente entre frontend y backend.

Estructura de Ramas

El repositorio seguirá un modelo de trabajo con tres ramas principales:

1. **main**: Rama principal y estable. Solo Sebastián puede aprobar *pull requests* hacia esta rama.
2. **feature/frontend**: Usada exclusivamente para el desarrollo de interfaces por Sebastián.
3. **feature/cleaning**: Usada por Steven para el desarrollo de scripts de limpieza y preprocesamiento del corpus.
4. **feature/core**: Rama donde Jorge trabajará integrando componentes, realizando pruebas y desarrollando funcionalidades generales.

Flujo de Trabajo y Ciclo de Desarrollo (M-LOOPS)

Se seguirá un ciclo **M-LOOPS** para organizar el desarrollo en iteraciones:

1. **Modelar** la tarea.
2. **Lograr** una implementación mínima.
3. **Observar** el funcionamiento.
4. **Optimizar** si es necesario.
5. **Probar** a fondo.
6. **Subir** los cambios como *pull request*.
7. **Merge** después de revisión por Sebastián.

Asignación de tareas

Tarea	**Responsable**
Desarrollo del frontend	Sebastián
Limpieza de corpus	Steven
Integración de módulos	Jorge
Automatización de scripts	Steven
Verificación y aprobación	Sebastián
Coordinación general	Jorge

División de Tareas Específicas para el Sistema de Recuperación de Información

- **1. Limpieza y Preparación del Corpus**

Responsable: Steven
Rama: feature/cleaning

- Recolección del corpus original.
- Eliminación de caracteres especiales, símbolos no útiles, y limpieza básica.
- Tokenización del texto.
- Conversión a minúsculas, eliminación de stopwords y lematización.
- Guardado del corpus limpio en carpeta corpus_limpio.

PROYECTO RI GRUPO 5

◇ 2. Indexación y Representación del Corpus

Responsable: Jorge
Rama: feature/core

- Creación del índice invertido.
- Representación vectorial de documentos (TF, TF-IDF).
- Implementación de estructuras de búsqueda eficientes.
- Almacenamiento de índices optimizados.

◇ 3. Interfaz de Usuario (Frontend)

Responsable: Sebastián
Rama: feature/frontend

- Maquetado con React + Tailwind.
- Desarrollo de componentes de búsqueda (barra de búsqueda, resultados, filtros).
- Visualización de resultados relevantes con puntuación.
- Conexión al backend para enviar consultas y mostrar resultados.

◇ 4. Módulo de Consulta y Ranking

Responsable: Jorge
Rama: feature/core

- Procesamiento de la consulta del usuario (normalización, tokenización, etc.).
- Búsqueda en el índice invertido.
- Cálculo de similitud (coseno, BM25 u otro).
- Ranking y retorno de los documentos más relevantes.

PROYECTO RI GRUPO 5

◇ 5. Evaluación y Métricas de Resultados

Responsable: Steven
Rama: feature/core

- Implementar métricas como *Precision*, *Recall*, *F1-score*.
- Comparación entre métodos de recuperación.
- Generación de informes para pruebas.

◇ 6. Integración y Pruebas Generales

Responsable: Sebastian
Rama: feature/core

- Pruebas unitarias y de integración.
- Revisión del flujo de datos entre frontend y backend.
- Simulación de búsquedas reales y casos extremos.

◇ 7. Aprobación y Control de Versiones

Responsable: Sebastián
Rol: *Maintainer*

- Revisión de *pull requests* de cada rama.
- Pruebas antes de hacer *merge* a main.
- Validación de calidad del código y coherencia del proyecto.

PROYECTO RI GRUPO 5

CODIGOS USADOS

Código de Preprocesamiento y Funcionalidades

preprocesamiento_corpus.py

Script encargado de la limpieza y preparación del corpus: tokenización, lematización, stopwords, etc.

GenerarCorpus.py

Script que genera un corpus a partir de archivos o fuentes de texto crudo, guardándolos en una estructura usable.

tfidf_api.py

Script que implementa la representación vectorial TF-IDF y su correspondiente API de consulta.

bm25_api_programmers.py

Implementación de un sistema de recuperación usando BM25, incluyendo ranking y búsqueda por relevancia.

eval_tfidf.py

Evaluación del modelo basado en TF-IDF usando métricas como precision, recall, F1.

eval_bm25_programmers.py

Evaluación del modelo BM25 con métricas similares a las del TF-IDF.

sistema_ri.py

Script principal que conecta los componentes del sistema de recuperación de información.

LINK REPOSITORIO:

https://github.com/SebasRo17/Proyecto-RI-1er-BIm