PROYECTOS FINALES

LISTA DE PROPUESTA DE PROYECTOS CON BASE A LOS TEMAS DE LA CURRICULA DE LA MATERIA

COMPLEJIDAD BAJA

- **Generador de árboles genealógicos**: Construir y manipular un árbol genealógico utilizando árboles binarios o multidireccionales. El proyecto debe permitir añadir miembros a la familia, encontrar antepasados comunes y mostrar los miembros de la familia en diferentes órdenes de recorrido.

- **Sistema de autocompletado:** Implementar un sistema de autocompletado utilizando una estructura de datos trie (árboles de búsqueda, donde las claves de búsqueda se encuentran en los nodos). Almacenar un diccionario de palabras en el trie y, a continuación, crear una funcionalidad para sugerir palabras completas en función de la entrada del usuario.

- **Navegador del sistema de archivos:** Simular un sistema de archivos utilizando una estructura de árbol. Implementar la funcionalidad para navegar a través de directorios, añadir/eliminar archivos o carpetas, y buscar archivos dentro del sistema utilizando diferentes estrategias de navegación.

COMPLEJIDAD MEDIA / ALTA

- **Simulación de un sistema de control de versiones (VCS):** Crear una versión simplificada de un VCS como Git, utilizando árboles (por ejemplo, árboles Merkle) para rastrear los cambios en los archivos a través de diferentes versiones. El sistema debería permitir a los usuarios confirmar cambios, volver a versiones anteriores y fusionar diferentes ramas, demostrando el recorrido de árboles y aplicar métodos de Hashing para comprobar la integridad del contenido de los archivos.

- **Indexación de bases de datos con B-Trees:** Implementar un sistema de indexación de bases de datos utilizando B-Trees para almacenar y recuperar de manera eficiente grandes conjuntos de datos. Hacer gestión de inserciones, eliminaciones y búsquedas, manteniendo las propiedades de un árbol B para lograr tiempos de acceso equilibrados. Este proyecto permite tener una visión práctica de cómo las bases de datos gestionan la indexación.

- **Grafo de red social con Hashing:** Desarrollar un modelo de red social donde cada usuario se represente como un nodo en un grafo y las conexiones (amistades) sean aristas. Utilizar mapas Hash para almacenar las relaciones y árboles de búsqueda binaria (BST) o árboles AVL para gestionar las publicaciones y actividades de cada usuario. El sistema debe permitir la búsqueda de usuarios, la recomendación de amigos (en función de las conexiones) y la determinación de rutas entre usuarios mediante algoritmos de recorrido en árbol.

- **Sistema de detección de plagios:** Implementar un sistema para detectar plagios entre documentos mediante el Hash de pequeños fragmentos de texto (utilizando técnicas como Rabin-Karp o rolling hash). Comparando los valores hash de los fragmentos de dos documentos, el sistema debe identificar regiones de similitud.

- **Motor de búsqueda de imágenes similares:** Desarrollar un motor de búsqueda de imágenes aplicando hashing perceptual para encontrar imágenes visualmente similares en una base de

datos. Al aplicar hashes a las imágenes en función de sus características visuales, el motor puede comparar rápidamente los hashes y devolver imágenes similares para una entrada determinada.

- **Shingling para la similitud de documentos:** Implementar un sistema para comparar documentos de gran tamaño (Big Data) utilizando shingling. Para ello, se dividen los documentos en pequeñas secuencias superpuestas (shingles) y, a continuación, aplicar funciones hash para representar cada shingle como una huella dactilar. El proyecto consistiría en comparar conjuntos de shingles hash para determinar la similitud entre documentos.

- **Coincidencia de secuencias de ADN con hashing:** Realizar hashing de secuencias de ADN para comparar similitudes genéticas. Utilizando técnicas como min-hashing o locality-sensitive hashing (LSH), el sistema identificaría regiones de similitud entre secuencias de ADN de forma eficiente, simulando cómo se procesan grandes bases de datos de información genética para su cotejo y análisis.

ASPECTOS GENERALES DE EVALUACIÓN DE LOS PROYECTOS

1. Comprensión de las estructuras de datos y su implementación
    - Evaluar su comprensión del uso adecuado de árboles, tablas hash y otras estructuras.
    - Justificación de por qué eligieron determinadas estructuras de datos en lugar de otras y cómo repercute su elección en el rendimiento de sus proyectos
2. Eficacia de la aplicación del algoritmo
    - Evaluar su capacidad para evaluar la complejidad en tiempo de sus algoritmos
    - Evaluar su capacidad para evaluar la complejidad de memoria de sus algoritmos
3. Optimización y análisis del algoritmo
    - Evaluar la implementación correcta de los algoritmos seleccionados
    - Evaluar su capacidad de optimización. Por ejemplo, si identifican que su función hash provoca demasiadas colisiones, intentar optimizarla o explicar posibles mejoras
4. Resolución y aplicación de problemas reales
    - Evaluar su capacidad para descomponer un problema en componentes manejables, aplicando algoritmos y estructuras de datos de forma eficaz
    - Evaluar si pueden trasladar los conocimientos teóricos a escenarios del mundo real
5. Prueba y depuración
    - Evaluar su capacidad para manejar y probar casos extremos; por ejemplo, instancias de entradas vacías, entras con gran cantidad de información, alta/baja similitudes en la entrada, etc.
    - Evaluar su capacidad para corrección de errores y excepciones en sus programas. Su capacidad para depurar eficazmente y manejar entradas no válidas.
6. Informe escrito y presentación
    - Evaluar que sean capaces de explicar claramente los algoritmos y estructuras de datos clave utilizados, tanto por escrito como oralmente.
    - Evaluar su capacidad de interpretar los resultados de sus pruebas, detectar y analizar cualquier discrepancia, así como, si son capaces de proponer posibles mejoras o trabajos futuros.

ICC006 – ANALISIS DE ALGORITMOS

LINEAMIENTOS DE EVALUACION DE PROYECTOS

Se realizarán cuatro avances de proyecto, cada entregable se realizará en su carpeta de DRIVE correspondiente, separando por subcarpetas cada avance

Avance 1: Definición del problema, planificación y diseño inicial (concepto y propuesta)
FECHA DE ENTREGA: 16 de septiembre 2024
Entregables
Definición del problema y objetivos:

- Definir claramente el problema que el proyecto pretende resolver.
- Proporcionar un breve resumen de la entrada, salida y comportamiento esperados del sistema.

Investigación preliminar y selección del algoritmo:

- Investigar problemas y algoritmos similares.
- Justificar la elección de las estructuras de datos (hashing, árboles o ambas) y por qué son apropiadas para el problema.
- Proporcionar un esquema de los algoritmos que se aplicarán, incluidas las complejidades de tiempo y espacio previstas.

Plan del proyecto:

- Presente un cronograma de actividades con el desglose de tareas para los próximos avances con responsable(s) cada una.

Avance 2: Desarrollo del algoritmo central y pruebas iniciales (prototipo)
FECHA DE ENTREGA: 09 de octubre 2024
Entregables
Implementación de la estructura de datos básica:

- Implementar la estructura de datos básica que se utilizará (por ejemplo, hashing o árbol).
- Para hashing: Implementar funciones hash y manejar colisiones.
- Para árboles: Implementar el recorrido, la inserción y la eliminación de árboles, si es necesario.
- Asegurando que operaciones básicas; por ejemplo, inserción, eliminación, búsqueda funcionan correctamente.

Desarrollo de algoritmos básicos:

- Desarrollar algoritmos básicos relacionados con la funcionalidad principal del proyecto (por ejemplo, detección de similitudes, recuperación de datos, indexación).
- Realización de pruebas básicas para garantizar que los algoritmos básicos funcionan como se espera con conjuntos de datos más pequeños.

Resultados preliminares y rendimiento:

- Proporcionar métricas de rendimiento iniciales (por ejemplo, complejidad temporal para insertar/buscar, tasa de colisión para hash).
- Identificar posibles problemas o cuellos de botella basándose en los resultados de las pruebas.

Plan del proyecto:

- Actualización del cronograma de actividades con el desglose de tareas para los próximos avances con responsable(s) cada una. Basándose de los resultados obtenidos hasta el momento.
- Minutas de las juntas (una semanal como mínimo) de equipo realizadas.

Avance 3: Funcionalidad completa, pruebas y optimización (versión alfa)
FECHA DE ENTREGA: 30 de octubre 2024
Entregables
Implementación completa:

- Completar la implementación de las funcionalidades restantes (por ejemplo, operaciones avanzadas, interfaz de usuario, manejo de datos).
- Implementar funciones adicionales como el manejo de conjuntos de datos más grandes, la optimización de la búsqueda o la adición de componentes de interacción con el usuario.

Pruebas exhaustivas:

- Probar el proyecto en diversos casos, incluidos casos extremos y pruebas de estrés (por ejemplo, grandes conjuntos de datos, alta similitud, condiciones extremas).
- Proporcionar resultados detallados de las pruebas, destacando los problemas encontrados durante las estas y cómo se resolvieron.

Optimización del rendimiento:

- Identificar las áreas de ineficiencia y optimizar (dentro de lo posible) el algoritmo o la estructura de datos (por ejemplo, mejore las funciones hash, equilibre los árboles).
- Proporcionar métricas de rendimiento actualizadas y compararlas con los resultados iniciales del Avance 2.

Gestión de errores y robustez:

- Asegurarse de que el sistema es robusto y gestiona los errores con eficiencia (por ejemplo, entrada no válida, datos que faltan).
- Implementar los mecanismos de gestión de excepciones (cómo validación de datos de entrada) o de emergencia que sean necesarios.

Plan del proyecto:

- Actualización del cronograma de actividades con el desglose de tareas para los próximos avances con responsable(s) cada una. Basándose de los resultados obtenidos hasta el momento.
- Minutas de las juntas (una semanal como mínimo) de equipo realizadas.

Avance 4: Pruebas finales, pulido y preparación de la presentación (versión beta)
FECHA DE ENTREGA: 13 de noviembre 2024
Entregables
Pruebas finales y validación:

- Realizar pruebas finales para validar la corrección, el rendimiento y la escalabilidad del proyecto.
- Asegurarse de que el sistema final funciona como se requiere y cumple todas las especificaciones del proyecto.

Documentación:

- Documentación completa del proyecto, incluidos comentarios del código, un manual de usuario y explicaciones detalladas de los algoritmos y estructuras de datos utilizados.
- Incluir una sección de análisis de rendimiento que explique cómo se escala el sistema y qué compensaciones se hicieron.
- Incluir una sección de conclusiones personales como grupales
- El documento debe tener una presentación formal. Con sus capítulos de Planteamiento y justificación del proyecto, Metodología, Experimentación y resultados, Conclusiones, Referencias

ICC006 – ANALISIS DE ALGORITMOS

Presentación final: Demostración y conclusión
FECHA DE ENTREGA: 25 de noviembre 2024

- Presentación de las funcionalidades clave y explicando cómo funciona el sistema.
- Presentar el problema, el enfoque, los algoritmos y los resultados de forma clara y concisa.
- Responder a las preguntas realizadas por compañeros o docente sobre la aplicación, los retos y los resultados del aprendizaje.