

Objetivo General

- Diseñar e implementar una aplicación para realizar búsquedas de texto en documentos.

Objetivos Específicos

- Implementar listas enlazadas y algunas variaciones
- Implementar árboles y algoritmos de ordenamiento
- Investigar y desarrollar una aplicación en el lenguaje de programación Java
- Investigar acerca de programación orientada a objetos en Java.

Descripción del Problema

Text Finder es una aplicación de escritorio escrita en Java que permite buscar texto en archivos .txt, .pdf y .docx. Cuando la aplicación indiza los documentos, el usuario puede buscar texto. La aplicación muestra los resultados de la búsqueda en todos los documentos y muestra al usuario extractos del documento donde aparece la palabra. El usuario puede ordenar los resultados por Nombre del archivo (Quicksort), fecha de creación (Bubblesort) y tamaño (Radixsort)

Las características principales de la aplicación son las siguientes:

1. *Administración de la biblioteca de documentos*: el usuario agrega documentos a la biblioteca. Puede especificar archivos individuales o carpetas con documentos. De igual forma, puede eliminar o actualizar documentos agregados a la biblioteca.
2. *Indización de la biblioteca*: la aplicación provee una opción para ejecutar la indización de los documentos. La indización consiste en lo siguiente:
 - a. Se parsea cada documento en la biblioteca.
 - b. Por cada palabra del documento, se inserta en un árbol AVL. Cada nodo del árbol tiene dos elementos: la palabra y una lista de ocurrencias. La lista de ocurrencias debe guardar el documento en el que aparece y alguna posición dentro del documento, de forma tal que cuando se busque la palabra, la aplicación pueda mostrar la palabra junto con texto alrededor.
3. *Búsquedas de texto*: el usuario puede ingresar un texto para buscar. Puede ingresar una frase y seleccionar buscar cada palabra individual o en frase. La aplicación muestra la palabra o la frase resaltada de un color junto con el texto alrededor. Muestra todas las

apariciones en todos los documentos indizados.

- a. El equipo de trabajo deberá diseñar alguna forma para que la búsqueda por frases sea precisa. Es decir, si se busca por frase y la frase es: "Costa Rica es mi país", los resultados solo deben incluir documentos con dicha frase exacta y no documentos que contengan las palabras individuales.
4. Abrir documento: El usuario puede abrir el documento en **la posición** donde aparecen las ocurrencias desde la aplicación.

Documentación requerida

Debe entregarse un documento PDF llamado "Documento de diseño" que incluya las siguientes secciones:

1. Introducción
2. Tabla de contenido
3. Breve descripción del problema
4. Descripción de la solución propuesta
5. Decisiones de diseño. Para cada decisión relevante:
 - a. Alternativas consideradas
 - b. Alternativa seleccionada y razones de la selección
6. Diagrama de clases UML de la solución propuesta (construido previo a la implementación)
7. Problemas encontrados
8. Preguntas abiertas

Deberá entregarse un documento PDF llamado "Planificación del proyecto" que contenga lo siguiente:

1. Lista de historias de usuario (pueden usar Azure DevOps o Jira para llevar la lista de Tareas, pero el documento debe encontrar la lista de estas)
2. Plan de iteraciones que agrupen cada bloque de historias de usuario por Sprint, de forma que se vea un desarrollo incremental. Se deberán de crear tres Sprints.
3. Asignación de tareas a cada miembro del equipo.

Aspectos operativos y evaluación

1. **Fecha de entrega: De acuerdo al cronograma del curso**
2. El proyecto tiene un valor de 20% de la nota del curso.
3. El trabajo es **en grupos de 4 personas**.
4. Es obligatorio utilizar un GitHub para el manejo de las versiones. Se debe evidenciar el uso de *commits* frecuentes.
5. Deben entregar en el TEC Digital un zip que contenga:
 - a. PDFs de los documentos
 - b. README.txt con el link al repo de Github
6. Es obligatorio integrar toda la solución, es decir, debe estar la UI y la lógica de negocios integrada.
7. La presentación funcional tendrá un valor de 70%, la documentación externa 15% y la documentación de diseño 15%.

8. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
9. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado, se recomienda que realicen la documentación conforme se implementa el código.
10. La documentación se revisará según el día de entrega en el cronograma.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial.
13. Aun cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
14. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
15. Si no se utiliza Git se obtiene una nota de 0.
16. Si la documentación no se entrega en la fecha indicada se obtiene una nota de 0.
17. Si el código no compila se obtendrá una nota de 0.
18. El código debe desarrollarse en Java, si no, se obtendrá una nota de 0.
19. La nota de la documentación debe ser acorde a la completitud del proyecto.
20. Si alguna persona integrante del proyecto no se presenta a la revisión se le asignará una nota de cero en la nota final del proyecto.
21. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
22. Cada grupo tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
23. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
24. Cada estudiante es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
25. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.