

Proyecto II – Text Finder

Instituto Tecnológico de Costa Rica
Área de Ingeniería en Computadores
Algoritmos y Estructuras de Datos I (CE 1103)
Primer Semestre 2019
Valor 20%



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.
- Aplicar **patrones de diseño** en la solución de un problema.
- Utilizar UML para modelar la solución de un problema.

Descripción del Problema

Text Finder es una aplicación de escritorio escrita en Java que permite buscar texto en archivos .txt, .pdf y .docx. Una vez que la aplicación ha indizado los documentos, el usuario puede realizar búsqueda de 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 previamente 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 los siguiente:
 - a. Se parsea cada documento en la biblioteca.
 - b. Por cada palabra del documento, se inserta en un árbol binario de búsqueda. 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.

4. Abrir documento: El usuario puede abrir el documento en la posición donde aparecen las ocurrencias desde la aplicación.

Documentación requerida

1. Internamente, el código se debe documentar utilizando JavaDocs.
2. Dado que el código se deberá mantener en GitHub, la documentación externa se hará en el Wiki de GitHub. El Wiki deberá incluir:
 - a. Breve descripción del problema
 - b. **Planificación y administración del proyecto:** se utilizará la parte de project management de GitHub para la administración de proyecto. Debe incluir:
 - Lista de features e historias de usuario identificados de la especificación
 - Distribución de historias de usuario por criticalidad
 - Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental
 - Descomposición de cada user story en tareas.
 - c. Diagrama de clases en formato JPEG o PNG
 - d. Descripción de las estructuras de datos desarrolladas.
 - e. Descripción detallada de los algoritmos desarrollados.
 - f. Problemas encontrados en forma de bugs de *github*: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.

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 parejas**.
4. Es obligatorio utilizar un GitHub.
5. Es obligatorio integrar toda la solución.
6. El código tendrá un valor total de 85% y la documentación 15%.
7. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
8. 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.
9. La nota del código NO podrá exceder en 35 puntos la nota de la documentación, por lo cual se recomienda documentar conforme se programa.
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. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - b. Si no se utiliza un manejador de código se obtiene una nota de 0.

- c. Si la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - d. Sí el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. El código debe ser desarrollado en Java, en caso contrario se obtendrá una nota de 0.
 - f. Si no se siguen las reglas del formato de email se obtendrá una nota de 0.
 - g. La nota de la documentación debe ser acorde a la completitud del proyecto.
14. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto. El único requerimiento que se consultará durante la defensa del proyecto es el diagrama de clases, documentación interna y la documentación en el manejador de código.
 15. Cada estudiante tendrá como máximo 15 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.
 16. 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.
 17. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.