

## Proyecto III – Cloud Drive

Instituto Tecnológico de Costa Rica  
Área Académica Ingeniería en Computadores  
CE2103 - Algoritmos y Estructuras de Datos II  
Primer Semestre 2022  
Valor: 20%



### Objetivo General

- Desarrollar una aplicación web utilizando React, NodeJS, MongoDB y C++

### Objetivos Específicos

- Aplicar conceptos de compresión
- Investigar y desarrollar una aplicación en el lenguaje de programación JS y C++
- Aplicar patrones de diseño en la solución de un problema.

### Descripción del Problema

Debe construir una solución de software que permita a los usuarios realizar las siguientes tareas:

1. Subir archivos y comprimirlos mediante Huffman, LZW, LZ77 y LZ78
2. Buscar y descargar archivos previamente subidos
3. Agregar tags que pueden ser utilizados para las búsquedas posteriores. Por ejemplo el usuario puede subir un archivo y posteriormente agregarle etiquetas como TEC, PROYECTO\_1 y utilizar dichas etiquetas para buscar
4. El sistema es multiusuario, por lo que debe proveer mecanismos de autenticación

### Componentes de la solución

#### Cliente Web

Es una aplicación web construida utilizando React. Debe proveer una interfaz gráfica sencilla para cumplir con los requerimientos. Esta aplicación necesita del REST API provisto por NodeJS para completar sus funciones.

#### REST API

Es una aplicación escrita en NodeJS que expone REST APIs que permiten a la aplicación web cumplir con las funcionalidades solicitadas por los usuarios. Debe diseñar correctamente los API y aplicar buenos principios de programación y organización de código.

Cuando la aplicación web sube un archivo, invoca un REST API y como parte de los parámetros, indica el algoritmo de compresión que debe utilizar. El REST API invoca una biblioteca de C++ que recibe los datos del archivo y el algoritmo de compresión. Dicha rutina se encarga de realizar la compresión del archivo. El archivo comprimido se almacena en una base de datos de MongoDB. Dicha base de datos también contiene información sobre el usuario al que pertenece el archivo y las etiquetas

#### Base de Datos

Los archivos, etiquetas, información de usuarios y cualquier otro dato, debe almacenarse en una base de datos NoSQL.

#### Puntos Extra

Si la solución completa se compila utilizando GitHub Actions/Azure DevOps y se despliega en Azure

(incluyendo autenticación con Azure Active Directory), se le darán 30 puntos más.

**Aspectos operativos y evaluación:**

1. **Fecha de entrega: De acuerdo con el cronograma del curso y lo establecido en el TEC Digital**
2. **Es obligatorio usar Linux para este proyecto**
3. **Cualquier estructura de datos como listas, deben ser implementadas por los estudiantes.**
4. El proyecto tiene un valor de 15% de la nota del curso.
5. El trabajo es en **grupos de tres persona.**
6. Deben entregar en el TEC Digital un documento con el link del repositorio de GitHub, Jira y el PDF de la documentación. En ambas herramientas deben dar acceso al correo del profesor.
7. Es obligatorio utilizar un Git y GitHub para el control de versiones del código fuente y evidenciar el uso de Commits frecuentes.
8. Es obligatorio integrar toda la solución.
9. El código tendrá un valor total de 70%, la documentación externa 20% y la defensa un 10%.
10. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
11. 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.
12. La nota de la documentación externa es proporcional a la completitud del proyecto.
13. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
14. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la primera cita de revisión oficial.
15. Aún cuando el código y la documentación externa tienen sus notas por separado, se aplican las siguientes restricciones
  - a. Si no se entrega documentación externa, automáticamente se obtiene una nota de cero en la nota final del proyecto.
  - b. Si no se utiliza un manejador de código se obtiene una nota de cero en la nota final del proyecto.
  - c. Si la documentación externa no se entrega en la fecha indicada se obtiene una nota de cero en la nota final del proyecto.
  - d. Si el código no compila se obtendrá una nota de cero en la nota final del proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
  - e. El código C++ debe ser desarrollado en Linux, en caso contrario se obtendrá una nota de cero en la nota final del proyecto.
16. La revisión de la documentación será realizada por parte del profesor. Podría ser revisada, por parte del profesor, antes o después de la cita de revisión del proyecto. Durante la defensa del proyecto sí se revisará el diagrama de clases, la documentación interna y la documentación en el manejador de código.
17. 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.
18. 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.
19. 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 al profesor para coordinar el préstamo de estos.

20. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.