Estructuras de Datos y Algoritmos

Jaime Andrés Torres Bermejo

Esteban Gonzalez Amaya

Sección 5 - Equipo 4

Laboratorio 1

### 3.1.5. Instrucciones para modificar la función para cargar archivos y consultar datos:

- ¿Cómo implementaría la función? ¿Qué orden de complejidad tendría la función implementada?

- Para encontrar la información requerida buscamos los criterios verificando que en la columna dada estuviese el criterio establecido. Esta función tendría un orden de crecimiento/ complejidad O(n).

### 3.2.2. Discusión en clase sobre GitHub, VS Code y Código implementado.

* ¿Cuál es el ciclo regular para actualizar código en un repositorio GIT?
* Una vez hecho algún cambio debemos hacer Git: Stage para añadir los cambios al Git: commit que haremos para notificar a todos los integrantes

del grupo las actualizaciones del código que hicimos. Para visualizar estos cambios los otros integrantes deben utilizar el comando Git: pull para posteriormente verificar que no haya conflictos y consolidar los cambios con la herramienta Git: push.

* ¿Qué ventajas y limitantes tiene el uso de Ramas/Branches?

Las principales ventajas del uso de ramas son la posibilidad de plantear varias soluciones a un

reto, trabajar basándose en los cambios recientes, así como probar el código sin afectar el proyecto final. La única desventaja que podríamos encontrar seria la cantidad de procesos y comandos requeridos para consolidar un cambio, sin embargo, una vez se esté familiarizado, la cantidad de requerimientos para realizar un cambio funcionan como barreras de seguridad para proteger el proyecto final.

* ¿Cuáles serían las buenas prácticas para solucionar conflictos?
* Para solucionar los conflictos debemos primero realizar Git: pull para ver qué cambios están generando el conflicto. Para finalmente decidir qué alteraciones deben permanecer podemos comparar las distintas versiones del código y tomar la decisión de mantener uno o más cambios o no conservar ninguno.
* ¿Qué orden de complejidad tendría las funciones (consulta y lectura de archivo)?
* El orden de complejidad que tienen la funciones implementas es O(n), donde la cantidad de datos aumentará la cantidad de tiempo que toma la función en responder.
* ¿Cómo podría reducir o aumentar la complejidad de la consulta?
* Para aumentar la complejidad podríamos añadir más iteraciones y revisar si todos los elementos coinciden aumentando la cantidad de veces que debemos revisar la información para encontrar datos coincidentes. Así mismo podríamos encontrar formas mas eficientes de buscar y encontrar los elementos que buscamos para reducir su complejidad, por ejemplo, utilizar columnas y filas para evitar revisar todos los datos hasta encontrar el requerido.
* ¿Cómo afecta un TAD en la complejidad?, ¿Qué alternativas existen?
* La manera en la que revisemos los datos afectará la complejidad temporal de la tarea. El objetivo siempre consistirá en encontrar la manera más óptima para buscar, identificar y mostrar los datos y coincidencias que se requieran. Las alternativas que existen son: Arboles binarios, listas y matrices.