Estructuras de datos y algoritmos – Laboratorio 1

Sección 7 Grupo 6

Gloria Guevara + Camilo García

**3.1.5**

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

Realizamos un recorrido inicial buscando el id de las películas que coincidieran con el director, y después otro recorrido buscando la fila en el otro archivo de datos que tuviera este mismo id. Una vez filtrada esta fila, se comparaba el valor de la calificación y se agregaba al counter dependiendo si era buena o no.

Asumiendo el peor escenario posible, en el que todas las películas son del mismo director y tienen calificación mayor o igual a 6, esto implica n\*n recorridos. Es decir que la función crecerá con orden exponencial cuadrático.

**3.2.2**

− ¿Cuál es el ciclo regular para actualizar código en un repositorio GIT?

Al escribir cada uno en su propia rama, se hace un proceso de pull, commit y push para irla actualizando en github. Al finalizar la edición en la rama, se hace merge con la rama principal (master).

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

Una desventaja es que el código no se actualiza al momento irlo editarlo desde las ramas y esto ocasiona que los colaboradores lleguen a escribir las mismas líneas y se genere error.

Una ventaja es que al guardarse el proceso podemos comparar el código editado con el de la rama master o original y así ver nuestro progreso.

− ¿Cuáles serían las buenas prácticas para solucionar conflictos?

Acodar previamente cual es el colaborador que editara determinadas funciones para así poder evitar conflictos a la hora de editar el código y esto ayudara a ganar tiempo. En caso de escribir la misma función comparar ambas para escoger la mas eficiente para la buena ejecución del programa.

Después de practicar y variar los parámetros de consultas de estructuras de datos considere y discuta las siguientes preguntas:

− ¿Qué orden de complejidad tendría las funciones (consulta y lectura de archivo)?

El orden de complejidad en las funciones para cargar los datos, o consultar dentro de solo un archivo base (sin hacer comparaciones entre ambos archivos), será lineal, ya que máximo se tendrá que hacer el mismo recorrido N un numero constante de veces. Para la función de consulta en la cual hay que buscar datos en ambos archivos csv, el orden de complejidad sería cuadrático, ya que hace un recorrido N dentro de otro recorrido N.

− ¿Cómo podría reducir o aumentar la complejidad de la consulta?

Para reducir la complejidad de consulta seria poner todos los datos a consultar en un mismo archivo csv y para aumentar la complejidad sera aumentar mas archivos en los cuales toque consultar la Informacion.

− ¿Cómo afecta un TAD en la complejidad?, ¿Qué alternativas existen?

Un TAD afecta la complejidad en sentido de que no tendríamos que estar recurriendo a todos los métodos que necesitamos para acceder a la información del archivo.

Una alternativa para reducir la complejidad seria inicializar el archivo como un elemento menos complejo de manejar.

En link de nuestro repositorio es:

https://github.com/EDA-SEC-07-EQUIPO-06/Lab1\_202020