

## DOCUMENTACIÓN TAREA 2

-DIA 1: En esta practica se pide algo similar que la anterior, pero en vez de pedir un valor el programa leera el fichero que esta dividido en bloques, también con formato tabular. Mi primera intención para dividir los diferentes bloques del fichero era usar "Split" al igual que en la practica anterior, pero en esta ocasion era mucho mas util y eficaz usar un Substring ya que te sirve para comparar caracteres en una posición concreta dentro de cada linea del fichero.

NOTA: para el titulo del documento se pide la fecha del dia en el que el equipo realiza la operación, y lo primero que se me ha venido a la cabeza es usar los comandos de linux, aunque no me parece que sea lo correcto y supongo que tendra que hacerse desde Java.

-DIA 2: En el día anterior conseguí mostrar todo el contenido del fichero por pantalla y encontrar cuantas veces aparece el patrón "CAB" que es el que marca la division de cada bloque.

El siguiente paso ha sido hacer una copia exacta del fichero en otro fichero diferente creado desde Java, el primer error que he tenido ha sido querer implementar esto directamente en la función, con lo cual lo puse en el main y me aparecieron más errores.

El primero fue que declaraba el "BufferedWriter" dentro de del bucle while y lo que ocurría era que solo escribia la ultima linea, con lo cual llegue a la conclusión de que se sobrescribia ese mismo "Buffered" con lo cual lo declare fuera del bucle while y ya si que me copiaba casi todas las lineas en el fichero, pero no todas, ya que me faltaba por cerrar el "BufferedWriter" con "bf close();", tras esto todo funcionaba perfecto.

-DIA 3: El proximo objetivo ha sido pasar todo lo anterior a la función pero he tenido varios problemas, el primero que dentro de la funcion afecta el while, con lo cual solo escribia la ultima linea, entonces en este momento no sabia muy bien como enfocararlo y pense en crear una funcion dependiente de la funcion anterior, pero el resultado era aun peor ya que no escribia nada.

En ese momento me quede un poco atascado pero finalmente encuentre por internet el poner dentro del "Buffered" cuando declaras el nombre del fichero poner "true" ya que lo que hace es guardar todo el contenido dentro del mismo fichero y no sobrescribir nada y asi no mostrar la última linea.

-DIA 4: El siguiente paso ha sido crear los ficheros necesarios cada vez que encuentre un "CAB" dentro de la linea del documento, como primera opción intente meterla dentro de la función que lee y cuenta el numero de veces que aparece "CAB" pero no conseguia hacerlo correctamente ya que el contador cada vez que se salia de la funcion se reseteaba con lo cual solo tenia el valor de 0 o 1, entonces solo me creaba un archivo.

Para solucionarlo cree una funcion la cual la declaro justo tras la anterior para y pasarle a esta como parametro el resultado de la anterior que es el numero de veces que aparece "CAB" entonces asi ya si me creaba los ficheros, y por untimo fue poner que cada vez que no sea una linea con la estructura "CAB" la escribiese en el fichero anteriormente creado ya que cada vez que aparecia "CAB" creaba un nuevo fichero con el valor del contador pasado como parametro. Por ultimo me quedaba nombre los ficheros con el nombre de anterior a la cabecera, que como en este caso era siempre de la misma longitud decidi usar un substring de nuevo, y para la fecha use algunas librerias y la función "get date()" junto con "get year", "get month" y "get day", todo esto dentro de la aplicación de java.

-DIA 5: En este dia solo me quedaba adaptar el anterior shell script a la tarea 2 y dar por finalizada esta.