**Historias Clínicas**

**(Reporte)**

Introducción:

Se nos asignó crear lotes de pruebas para encontrar los posibles errores en ocho ejecutables para un hospital, cuya función era recibir una lista de historiales clínicos y ordenarlos ascendientemente.

Desarrollo e ideas:

El primer pensamiento que tuvimos fue someter a cada ejecutable a una lista de historias clínicas normal (como el ejemplo dado en las consignas), para así darnos cuenta de que programas tenían errores que podían ser detectados a simple vista.

Luego empezamos a pensar en casos más puntuales, como por ejemplo la lectura del archivo, si lo leía bien, o si no. Es clave tener casos que apunten a estos temas ya que de haber inconvenientes en la lectura o escritura del archivo, nunca se va a poder ordenar correctamente.

En este punto ya habíamos encontrado varios errores y decidimos taclear temas como los tipos de dato que se estaban usando en cada programa. Ya que el hospital tenía más de cuarenta mil historias clínicas por año, nos pareció pertinente crear un lote de prueba que nos permite verificar el uso de una variable Long en vez de int. También, al saber la gran cantidad de historias clínicas que puede haber, creamos un programa que produjera un archivo de alrededor de 50000 registros, para probar el tiempo de ejecución para un archivo tan grande.

Por último se nos ocurrieron casos como, que pasaría si hay dos pacientes con el mismo nombre, si no hubiera ningún paciente y casos que nos permitieran redondear nuestras ideas para poder sacar conclusiones.

Resumen:

* **hClinicas\_1.exec:** Al parecer este programa usa dos vectores de ordenamiento, uno para las historias clínicas y otro para los nombres. El vector de historias clínicas estaría ordenándose bien, el problema ocurre con el vector de nombres, ya que siempre se repite el primer nombre que aparece en el archivo, como si hubiera algún problema con el índice del vector. Posible solución: volver a mirar el algoritmo de ordenamiento del vector de nombres.
* **hClinicas\_2.exec:** Casi un programa sin problemas, hasta que llega al final. Este ejecutable ordena todo el archivo bien, pero mantiene la posición de la última entrada en la última posición de la salida. Posible solución: puede que el ciclo donde se imprime en el archivo de salida termine antes del largo del vector y que haya una sentencia de más afuera del mismo.
* **hClinicas\_3.exec:** Este programa también es casi perfecto. El único problema que se le encuentra es que no funciona para variables Long. Solución: usar Long en vez de int.
* **hClinicas\_4.exec:** Este parecería ser el mejor programa. Responde bien a todos los casos de prueba, aunque tarda una eternidad en resolver el caso fatiga. Ver la forma de hacer el algoritmo más eficiente.
* **hClinicas\_5.exec:** Este problema parecería andar bien… ¡Si solo se le pide ordenar un registro! Para más de un registro este programa deja de funcionar. Posible solución: lo más probable es que haya algún error en la escritura del archivo out.
* **hClinicas\_6.exec:** Este programa sufre de un error en la ejecución únicamente cuando se tiene la historia clínica con menor valor en la última posición. Posible solución: el problema debe estar en la escritura del archivo.
* **hClinicas\_7:** Este programa solo funciona correctamente para historias clínicas menores o iguales a nueve, de no ser así, la escritura del archivo es un desastre. Posible solución: puede que se esté leyendo el valor de la historia clínica como una cadena y se esté almacenando en un vector de una posición. ¡Usen int o Long!
* **hClinicas\_8.exec:** El programa ordena a partir de la segunda historia clínica, la primera queda en su lugar. Posible solución: puede que haya alguna sentencia de escritura fuera del ciclo que se usa.