**Tarea individual: Act-Integradora-1 Conceptos básicos y algoritmos fundamentales**

Arturo | Matricula

Logo, company name

Description automatically generatedGrant Keegan | A01700753

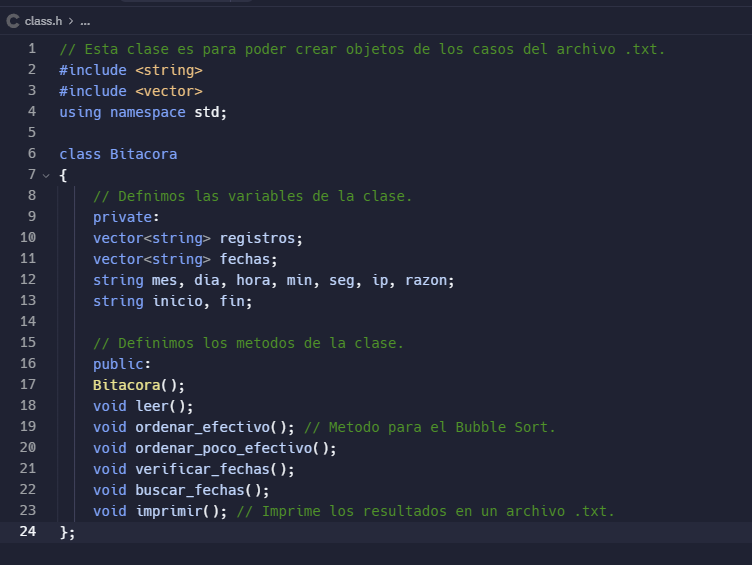
27 de marzo del 2023

Programación de estructuras de datos y algoritmos fundamentales (Gpo 850)

Profesor: Dr. Eduardo Arturo Rodríguez Tello

En esta actividad, nuestro objetivo es escribir un programa en C++ que tome una lista de 16,000 entradas en un archivo .txt., por medio de algoritmos de ordenamiento (Bubble e insertion sort) ordene esas entradas por fecha, e imprima los resultados en otro archivo .txt.

**Parte 1: Diseñando la clase**

En esta actividad, lo primero que decidimos hacer fue diseñar nuestra clase con la que crearemos las instancias de objetos sobre las entradas del documento bitácora .txt. Creamos la clase ‘Bitácora’ en la que nuestros atributos fueron los 7 datos de los casos de errores (mes, día, hora, minuto, segundo, la dirección IP y la razón del problema).

El objetivo ahora de la función es tomar las mas de 16,000 lineas de texto, y usando la clase crear instancias para poder agregar a una lista, con los atributos respectivos. A partir de aquí podremos ordenar las entradas.

**Parte 2: Escribiendo los algoritmos.**

La actividad especifica que debemos de crear 2 diferentes tipos de algoritmos para resolver el mismo problema. Por lo que decidimos elegir los algoritmos de insertion y bubble sort para trabajar con los objetos creados por la clase en base al archivo.txt.

**Parte 3: Solicitando al usuario un intervalo de búsqueda**

En esta parte, fue necesario crear una función para que el usuario introduzca 2 fechas de inicio y fin de la búsqueda. Por ejemplo, del 16 de mayo del 2016 al 23 de junio del 2018, y nuestro programa solo imprima los elementos de la lista dentro de ese intervalo. Para esto fue necesario hacer varias cosas, comenzando con crear dos strings de inicio y fin en nuestra bitácora.

Text

Description automatically generated

Después de eso, debimos de escribir una forma para que el programa identifique los elementos de la lista **ya ordenada** y excluya los elementos de esta en la impresión.

**Parte 4: Imprimiendo la lista ordenada en un nuevo archivo .txt**

En esta ultima sección, lo que tenemos que hacer es imprimir los nuevos resultados de nuestro algoritmo ordenado en un nuevo archvo.txt, al que nombraremos bitacora\_ordenada. A este archivo se imprimirán incluso si el usuario definió los parámetros de que fechas se iban a seleccionar.

**Investigación y reflexión | Arturo**

…

**Investigación y reflexión | Grant**

Los algoritmos de ordenamiento son de los mas importantes que puede utilizar un programador, ya que se utilizan bastante en todos lados. Pero tenemos varias opciones para utilizar con estos algoritmos, y en diferentes casos debemos utilizar varios

En un video, aprendí las diferentes ventajas y desventajas que tienen distintos algoritmos de ordenamiento. Algunos funcionan mejor en listas de datos mas grandes. Una de las formas más efectivas que nosotros como programadores podemos elegir el mejor algoritmo de ordenamiento, es fijándonos en la complejidad de este. Usualmente los que tienen una de O(n^2) o mayor son menos efectivos cuando se lidiar con listas grandes tales como la de esta tarea.

**Notas y Referencias**

[1] https://www.youtube.com/watch?v=RfXt\_qHDEPw