Jorge Gavidia C.I. 18836874 Secc. C1

Planteamiento del problema:

Se debe emular un Twitter que es un servicio de microblogging, bajo varios parámetros q se describirán posteriormente.

Dado un archivo entrada.in que está compuesto por 3 secciones debe leerse y cargarse en una estructuras explicadas posteriormente, para proceder al manejo de dicha información.

Cuando el archivo esté cargado, se debe emular el inicio de sesión o añadir un nuevo usuario. Después de manejar estos datos, se procederá a guardarlos en un archivo salida.txt

Análisis:

El problema se debe hacer bajo el esquema del uso de listas simplemente y doblemente enlazadas, ya que vamos a añadir y eliminar datos de forma dinámica durante la ejecución del programa.

Explicación de la solución:

Estructuras a utilizar:

Listas simplemente enlazadas: Se utiliza esta estructura ya que no sabemos exactamente cuántos miembros de un tipo de datos vamos a almacenar**,** utilizando estas estructuras dinámicas, puede crecer y decrecer el número de objetos en el tiempo de ejecución.

Clase lista_twitter: esta es la clase principal donde se guardaran todos los datos de los usuarios cargados del archivo. Esta estructurada por nodos que poseen los siguientes campos: del tipo string nombre guarda nombre del usuario; del tipo string email guarda el correo de usuario; del tipo string usuario guarda el nombre de usuario a utilizar dentro del programa; del tipo string password guarda la contraseña del usuario; del tipo lista_seguidores se guardara en lista_seg la lista de seguidores; del tipo lista_siguiendo se guardara en lista_sig la lista de los siguiendo por el usuario; y por ultimo un nodoinfousurio que apunta a próximo. La adición se hace en forma ordenada alfabéticamente.

Clase lista_siguiendo y clase lista_seguidores: son clases similares, estas están ubicada en cada nodo de la clase lista_twitter para guardar a los siguiendo y seguidores de cada uno de los usuarios. La adición se hace en forma ordenada alfabéticamente.

Listas Doblemente enlazadas: Se utiliza esta estructura ya que no sabemos exactamente cuántos miembros de un tipo de datos vamos a almacenar**,** utilizando estas estructuras dinámicas, puede crecer y decrecer el número de objetos en el tiempo de ejecución y posee en su nodo un apuntador a siguiente y apuntador a anterior lo cual puede hacer 2 recorridos de adelante hacia atrás o el opuesto a este.

Clase lista_twetts: esta es la clase donde se guardaran todos los tweets de los usuarios cargados del archivo, en su nodo posee los siguientes campos: de tipo string tweet aca guarda el tweet leido; de tipo string usuario guarda el usuario que hizo el tweet; del tipo nodo_tweet *próximo un apuntador al siguiente nodo; de tipo nodo_tweet *anteriorun apuntador al nodo anterior. Esta lista se comporta como una pila en su inserción, siempre se hace al principio y dejando al más antiguo de ultimo. La adición por lo anterior expuesto se hace en forma ordenada cronológicamente.

Métodos utilizados para la resolución del problema:

-Carga del Archivo entrada.in:

Para este módulo del programa nos apoyamos en la librería fstream de c++ para abrir el archivo con ifstream, leer una línea completa con getline, y cerrar archivo con close. Ya podemos leer del archivo los datos.

Método cargar_archivo: leeremos línea a línea el archivo de entrada, buscando cada una de las secciones del archivo. Para esto implementamos 3 constantes que hacen referencia a cada una de las partes del archivo y la vamos buscando, si la conseguimos añadimos una bandera para que las líneas posteriores sean tratadas de la forma de ese fragmento de archivo.

Método usuarioss: recibe una línea del archivo correspondiente a Usuarios, en el método desglosa nombre, correo, usuario, contraseña para ser agregado a la lista usuarios.

Método siguiendoss: recibe una línea del archivo correspondiente a dos usuarios uno que sigue y otro que es seguido, se desglosa estos usuarios y son enviados a buscar_insertar_sig y a buscar_insertar_seg si estos dos usuarios no son el mismo.

Métodos buscar_insertar_sig y buscar_insertar_seg reciben dos usuarios uno que es seguido y el otro que es seguidor, dependiendo de cada uno de roles desempeñados se almacena en la lista opuesta de cada uno de los dos usuarios que se ven aca.

Método cargar_tweets: carga los tweets en una lista doblemente enlazada, recibe el tweet y el usuario que lo emitió y se carga de primero.

Menú Principal: Alli se dan 3 opciones registrar usuarios, iniciar sesión o salir.

Si opción es Registrar usuario:

Método agregar_usuarios(): se pide la información del usuario y se procede a llamar al método agregar y se agrega el usuario.

Si opción es iniciar sesión:

Método t.loguin(): Este método devuelve un puntero al usuario que desea iniciar sesión. Se lee el usuario y el password y se envía a buscar_loguin ();

Método buscar_loguin: recibe el usuario y el password, busca dentro de los objetos de la lista twitter si hay un objeto que cumple con esas condiciones. Si lo encuentra retorna el puntero a ese nodo, sino retorna null .

Método menu_loguin: si ya se pudo loguear puede entrar en este método, se muestran las diferentes funciones que hace el programa para el manejo de la cuenta.

Opción 1 Ver seguidores: Se lee un usuario y se ve la lista de seguidores de ese usuario.

Opción 2 Ver siguiendo: Se lee un usuario y se ve la lista de siguiendos de ese usuario.

Opcion 3 Crear Tweet: se pide que se ingrese el tweet, solo se admiten 140 caracteres, y se envía al método agregar de lista_tweets.

Opción 4 Eliminar Tweet: Se listan los tweets del usuario, que son los único que el puede eliminar, una ves escogido el tweet se guarda el apuntador de ese tweet, y se procede a eliminar el tweet en la lista .

Opción 5 Seguir: Se listan los usuarios a seguir y se guarda el puntero del elegido. Se busca dentro de la lista siguiendo para ver si ya se sigue, si es positivo devuelve el mensaje que ya sigue a este usuario, en caso contrario, procede a agregar este usuario a lista de siguiendo con el método buscar_insertar_sig y agregar al usuario seguido con buscar_insertar_seg. Pasando a ambos el nombre de usuario y el que queremos seguir.

Opción 6 Dejar de Seguir Usuario: Se listan los usuarios a seguir y se guarda el puntero del elegido. Se busca dentro de la lista siguiendo para ver si ya se sigue, si es positivo se elimina de la lista, y después se procede a eliminar el seguido del usuario que deje de seguir.

Opción 7 Ver muro: Muestra todos los tweets en forma descendente desde el mas antiguo hasta el más reciente.

Opción 8 Guardar Archivo: Apoyándonos en la librería fstream de c++ para abrir el archivo de salida con ofstream, fs para imprimir en el archivo, y close para cerrar el archivo.

Escribimos usuarios, y procedemos a leer la lista Lista_twitter, que posee los datos de cada uno de los usuarios, estos se aglomeran en la variable línea, y se imprimen cada usuario con sus datos

Escribimos siguiendos, y procedemos a leer la lista Lista_twitter, que posee los datos de cada uno de los usuarios, estos se aglomeran en la variable línea usuario y siguiendo, y se imprimen cada usuario con sus datos.

Escribimos tweets, y procedemos a leer la lista Lista_tweets, que posee los datos de cada uno de los usuarios y sus tweet, estos se aglomeran en la variable línea usuario y tweet, y se imprimen cada usuario con sus datos. Esta la imprimimos de una manera especial ya que el tweet más antiguo lo imprimimos de primero y el último en el último lugar.

Opción 9: Salir: acá salimos del usuario del cual estamos logueado.

Si opción Salir: Sale del programa.

Consideraciones del programa:

- Las palabras que hacen referencia las 3 partes del archivo, ¨usuarios, siguiendos, tweets¨, en el archivo de lectura debe estar escrito en minúscula.
- El manejo de Todo el archivo se hace con letras minúsculas.
- La salida del archivo se hace en letras totalmente minúsculas y lleva el nombre salida.in, llevando la extensión del archivo de entrada.
- En el muro, el usuario ve Todos los tweets. (Se consideró esta opción como válida en la página de la materia).