Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI)

Actividad de aprendizaje 10:
La Lista, implementación dinámica simplemente ligada

Víctor Agustín Díaz Méndez
Ingeniería en Informática

Estructura de Datos I (Sección D12)
Profesor: Dr. Gutierrez Hernandez Alfredo

Problema

Tome el problema de la actividad 06 (sólo con búsqueda lineal), y cubra las necesidades utilizando una lista simplemente ligada en lugar de una lista estática.

Requerimientos:

a) El estilo de programación debe ser Orientado a Objetos.

b) Considere de forma separada la clase Nodo respecto de la clase que servirá para instanciar los datos que almacena la lista.

c) La clase *Lista* y todas sus operaciones deberán alojarse en una librería, separándola de resto del programa.

d) El uso de la Lista debe hacerse exclusivamente a través de sus métodos.

e) Las operaciones a implementar, independientemente de que sean implementadas o no en éste programa son: inicializa, vacía, llena, insertar, elimina, recupera, busca (lineal), primero, ultimo, anterior, siguiente y anula;

Entregables:

1. Caratula (Nombre de la actividad y datos del alumno).
2. Resumen personal del trabajo realizado, y forma en que fue abordado el problema.
3. Código fuente.
4. Impresiones de pantalla que muestren la ejecución satisfactoria del programa.

Resumen:

La clase *Node*

En apariencia la implementación de la clase *Node* parece sencilla, y en si puede serlo pero ha que ser muy cuidadosos con ella. Primero ha que tomar en cuenta que hacemos uso punteros, lo que puede llegar a ser confuso o es muy sencillo cometer un error de sintaxis como usar o no usar asterisco cuando se debe.

Uno de los errores mas graves que cometí en el proyecto fue eliminado los espacios de memoria de los apuntadores *next* y *prev*, lo que provocaba que cuando quería acceder al nodo anterior o siguiente después de haber eliminado este me provocara error al intentar a un espacio de memoria no reservado. Finalmente me di cuenta de eso y el destructor de la clase solo elimina el espacio de memoria de *data.*

La clase *List*

Para los proyectos anteriores utilice un lista estática que solo podía manejar un tipo de datos. Así que decidí implementar de una vez una Lista dinámica que hace uso de *templates*.

Hubo dos funciones que decidí no implementar en el programa a pesar de que en los requerimientos esta especificado que se haga: *inicializa* y *llena.* Tuve mis motivos para hacer eso, para *inicializa* porque haría prácticamente lo mismo que el constructor, y para el caso de *llena* es porque no tiene sentido alguno ponerlo (al menos que existiera algún detalle que no haya visualizado) porque la lista es dinámica y en teoría no se puede llenar (a menos que nos acabáramos la memoria, cosa que seria muy rara).

Para el constructor solo especifique que igualara el ancla a *nullptr.*

El constructor copia hace un recorrido de la lista a copiar y cada elemento lo inserta en la lista en la que se copian.

El destructor solo llama al método *deleteAll.*

El método *isEmpty* solo compara el ancla con *nullptr*;

InsertData es un método un poco más complicado que los anteriores mencionados porque se tiene que verificar que la dirección de memoria en donde se insertara es valida y si es así tiene que lanzar una excepción. También tiene que verificar que se haya podido reservar el espacio de memoria para el nuevo nodo y sí no se puede lanzar otra excepción. Hay dos tipos de posiciones donde se puede insertar al principio y cualquier otra. Si se inserta al principio nuestro nuevo nodo a puntara a la misma dirección que el ancla, y el ancla apuntara al nuevo nodo. Al insertar en cualquier otra posición, el nuevo nodo apuntara al donde apunta el nodo de la posición a insertar, para que después el nodo donde se insertar apunte al nuevo nodo.

Eliminar es mas sencillo, el nodo anterior del que se eliminara apuntara apunta el nodo a eliminar. Después se libera su espacio de memoria.

getNextPos verifica si la posición es valida devuelve la posición a donde apunta el nodo.

getPrevPos verifica si la posición es valida, hace un recorrido de la lista hasta encontrar un nodo que apunte a la posición, si es así devuelve la posición del nodo, si no termina cuando algun nodo apunte a *nullptr* y devuelve *nullptr*.

getFirstPos devuelve el ancla.

getLastPos revisa si la lista esta vacía*,* sí no recorre toda la lista hasta encontrar el ultimo elemento y lo retorna.

findData revisa si la lista esta vacia, sí no recorre la lista hasta encontrar un nodo que contenga una data igual a la buscada.

retrieve revisa si la posición es valida, si lo es obtiene la *data* del nodo.

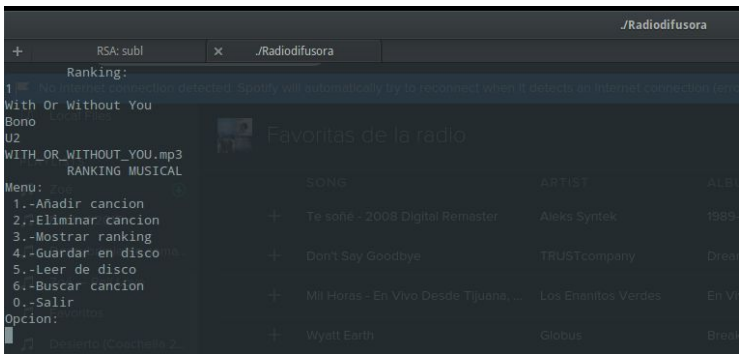
printData recorre la lista e imprime la *data* de los nodos.

deleteAll recorre la lista haciendo que un nodo auxiliar guarde los espacios de memoria del ancla mientras esta cambia al siguiente nodo, después se elimina la dirección de auxiliar, el proceso se repita hasta que ancla sea igual a *nullptr.*

Leer y escribir de disco funciona prácticamente igual pues funciona usando otros métodos de la clase. Solo cambia la manera en al que recorre el arreglo.

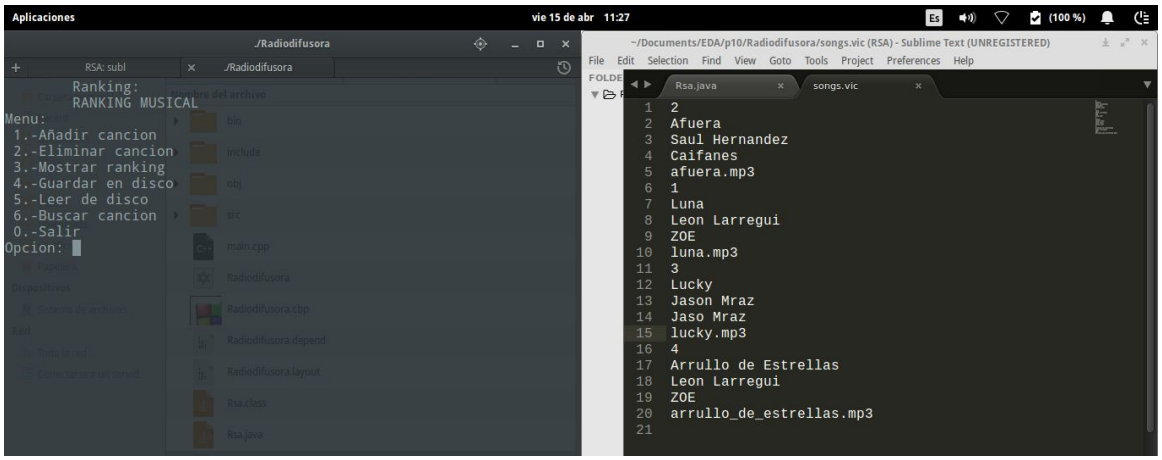
Ejecución satisfactoria:

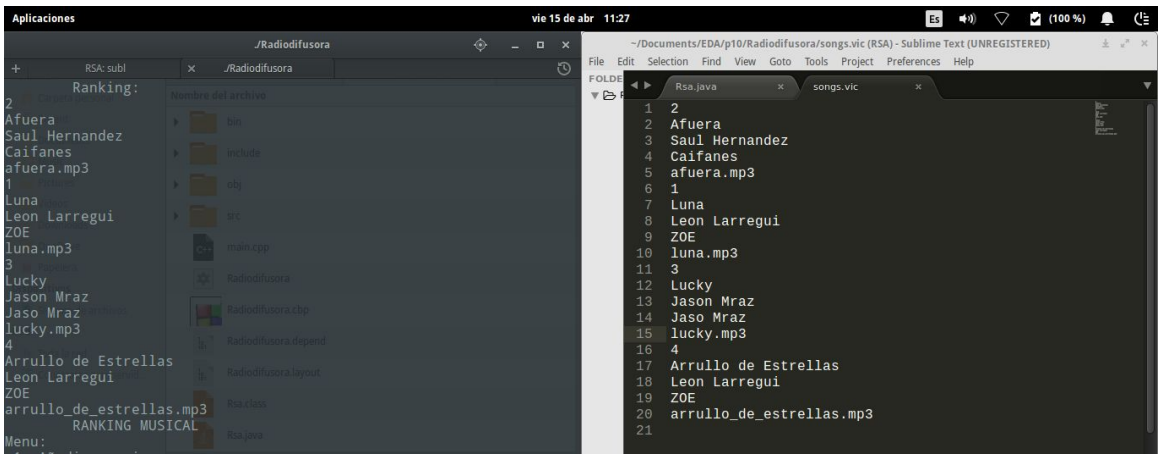
Inserción



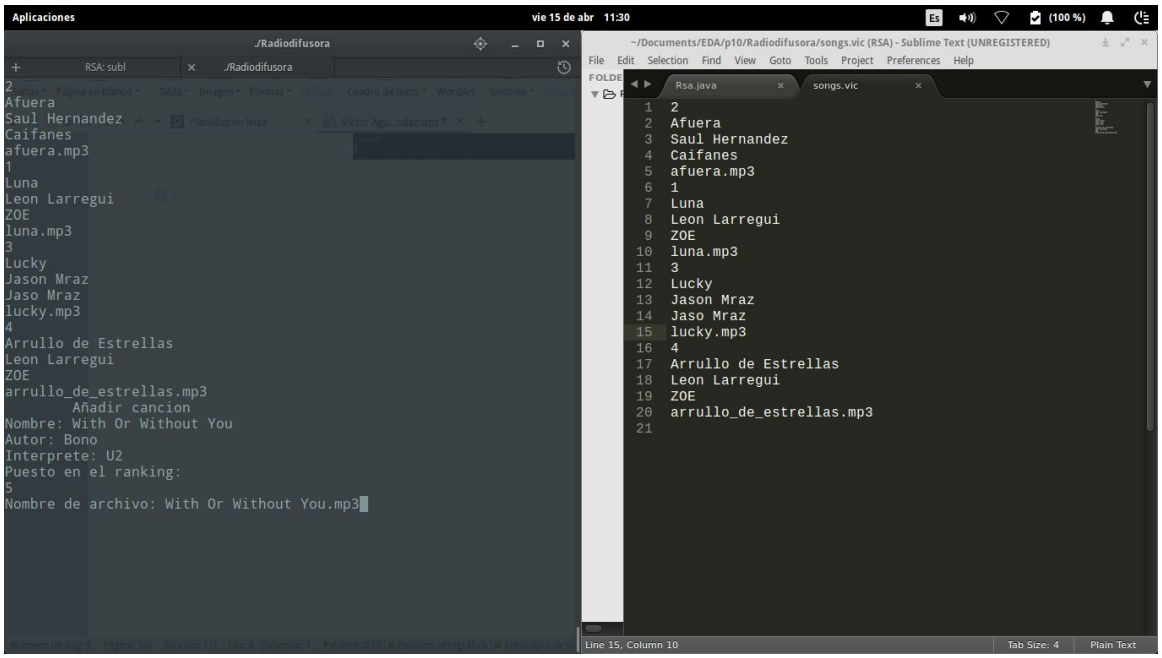
Buscar

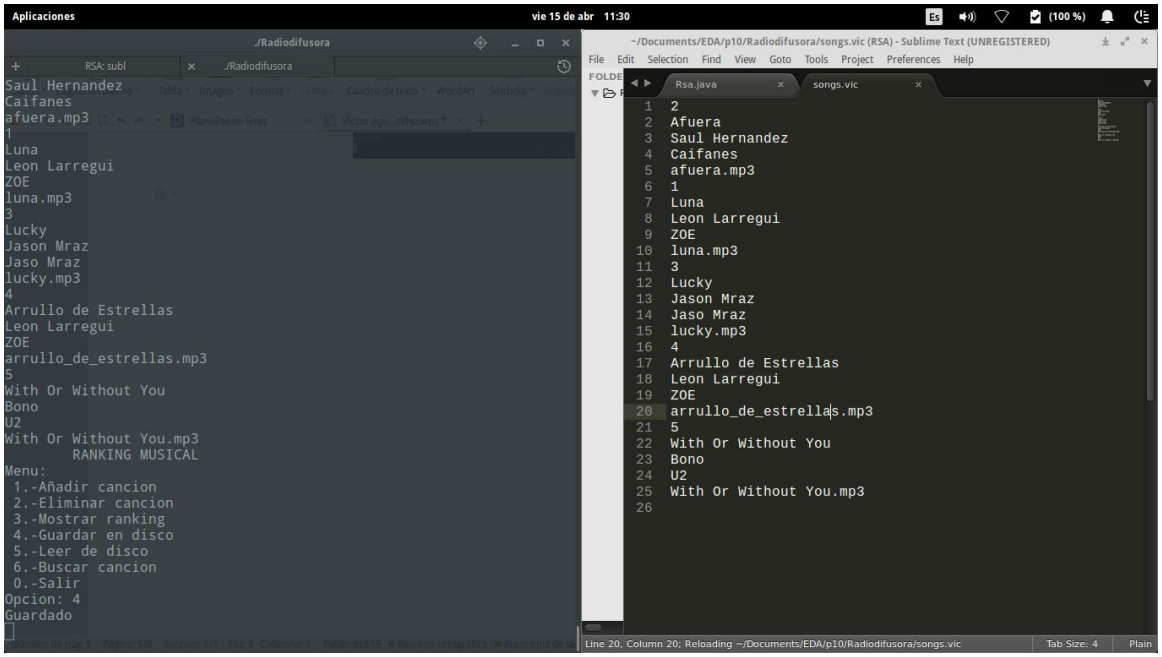
Lectura de disco





Guardar


Aplicaciones
vie 15 de abr 11:30
./Radiodifusora
RSA: subl
./Radiodifusora
2
Afuera
Saul Hernandez
Caifanes
afuera.mp3
1
Luna
Leon Larregui
ZOE
luna.mp3
3
Lucky
Jason Mraz
Jaso Mraz
lucky.mp3
4
Arrullo de Estrellas
Leon Larregui
ZOE
arrullo_de_estrellas.mp3
Añadir cancion
Nombre: With Or Without You
Autor: Bono
Interprete: U2
Puesto en el ranking:
5
Nombre de archivo: With Or Without You.mp3
~/Documents/EDA/p10/Radiodifusora/songs.vic (RSA) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rsa.java
songs.vic
Line 15, Column 10
Tab Size: 4
Plain Text


Aplicaciones
vie 15 de abr 11:30
./Radiodifusora
RSA: subl
./Radiodifusora
Saul Hernandez
Caifanes
afuera.mp3
1
Luna
Leon Larregui
ZOE
luna.mp3
3
Lucky
Jason Mraz
Jaso Mraz
lucky.mp3
4
Arrullo de Estrellas
Leon Larregui
ZOE
arrullo_de_estrellas.mp3
5
With Or Without You
Bono
U2
With Or Without You.mp3
RANKING MUSICAL
Menu:
1.-Añadir cancion
2.-Eliminar cancion
3.-Mostrar ranking
4.-Guardar en disco
5.-Leer de disco
6.-Buscar cancion
0.-Salir
Opcion: 4
Guardado
~/Documents/EDA/p10/Radiodifusora/songs.vic (RSA) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Rsa.java
songs.vic
Line 20, Column 20; Reloading ~/Documents/EDA/p10/Radiodifusora/songs.vic
Tab Size: 4
Plain