



## Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI)

### Actividad de aprendizaje 11: La Lista, implementación dinámica doblemente 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 10, y cubra las necesidades utilizando una lista doblemente ligada en lugar de una simplemente ligada.

## Requerimientos:

- a) El estilo de programación debe ser Orientado a Objetos.
- b) Debe ser suficiente con el cambio de librería para que el resto de l programa funcione de forma idéntica.

## 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 Lista Circular Doblemente Ligada Con Encabezado Dummy**

Abordar el tema fue sencillo una vez teniendo resuelto el problema anterior, solo tenia que cambiar un poco la forma en la que manejaba la lista. EL primer factor a considerar es que los nodos ya guardan direcciones al nodo anterior lo que ayuda bastante a eficientar el programa ya que no es necesario recorrer la lista para obtener el atributo anterior. También tenemos la ventaja de en lugar de manejar la lista directamente con un ancla tenemos dentro insertado un nodo que nos sirve para manejar la lista mas fácilmente, a este nodo lo llamamos *encabezado dummy*.

La mayoría de los métodos se volvieron mas sencillos de implementar como *insertar* donde ya no tenemos que hacer distintos procedimientos si se va a insertar al principio, o si la lista esta vacía. Esto es porque prácticamente todas las inserciones son bajo el mismo estado porque la lista nunca esta vacía (porque tiene un nodo *dummy*) y no hay ultimo elemento porque el diseño es circular.

Otra ventaja es que para los métodos *obtener anterior* y *obtener siguiente* es muy sencillo obtener su correspondiente elemento, pues las direcciones de ambos están guardadas en los atributos del nodo.

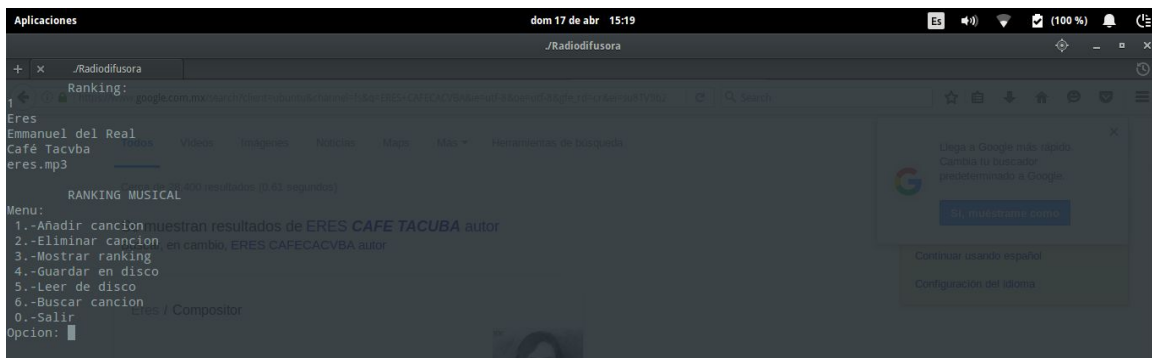
El mismo caso es para obtener el primer y ultimo elemento de la lista, solo es necesario obtener el siguiente del encabezado o el anterior.



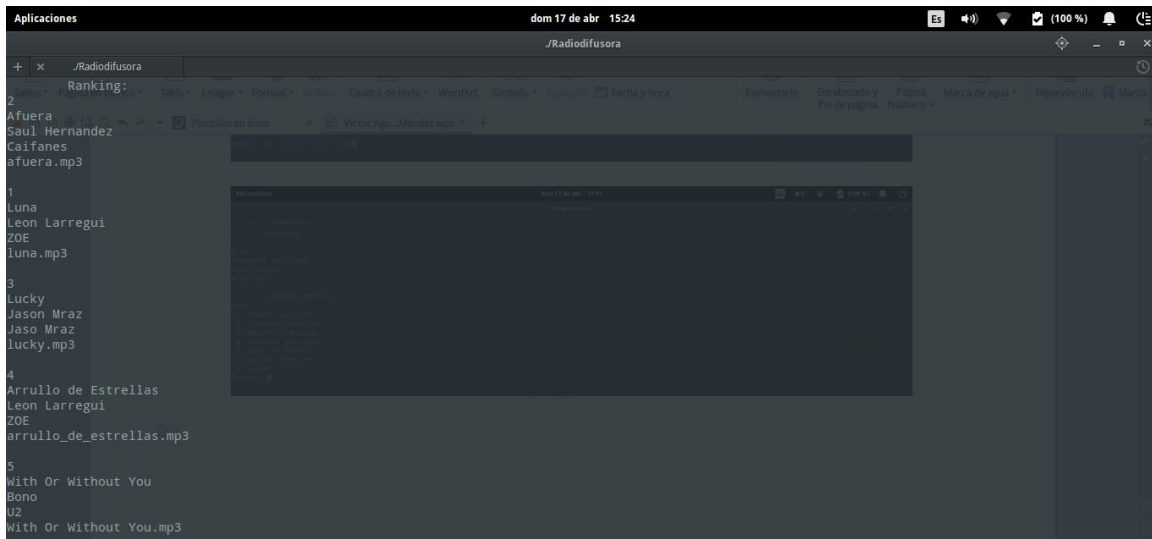
Para recorrer la lista la única diferencia que habría sería que en lugar de detenernos cuando la lista apunte a donde mismo apunta el ancla se detendrá al llegar al encabezado.



## Ejecución satisfactoria: Inserción



## Leer de disco





## Eliminar

Nombre de la canción a eliminar  
With Or Without You

```
Aplicaciones dom 17 de abr 15:27
J/Radiodifusora
+ x J/Radiodifusora
Ranking:
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
Universidad de Guadalajara
```

## Búsqueda

```
Aplicaciones dom 17 de abr 15:30
J/Radiodifusora
+ x J/Radiodifusora
Busqueda por nombre
Nombre de la canción: Luna
Canción encontrada
1 Luna
2 Leon Larregui
3 ZOE
4 luna.mp3
showSong() void
try {
    s = songList.retrieve(songList.findData(s));
} catch (ListException ex) {
    cout << "No se encontro" << endl;
    cin.get();
}
```

## Guardar en disco

```
Aplicaciones dom 17 de abr 15:32
~/Documents/EDA/p11/Radiodifusora/songs.vic (RSA) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
songs.vic - p10/Radiodifusora lista.h songs.vic - p11/Radiodifusora
1
2 Afuera
3 Saul Hernandez
4 Caifanes
5 afuera.mp3
6 1
7 Luna
8 Leon Larregui
9 ZOE
10 luna.mp3
11 3
12 Lucky
13 Jason Mraz
14 Jaso Mraz
15 lucky.mp3
16 4
17 Arrullo de Estrellas
18 Leon Larregui
19 ZOE
20 arrullo_de_estrellas.mp3
21

J/Radiodifusora
+ x J/Radiodifusora
Arrullo de Estrellas
Leon Larregui
ZOE
arrullo_de_estrellas.mp3

RANKING MUSICAL

Menu:
1.-Añadir canción
2.-Eliminar canción
3.-Mostrar ranking
4.-Guardar en disco
5.-Leer de disco
6.-Buscar canción
0.-Salir
Opcion: 4
Guardado
```