	<b>Carátula para entrega de prácticas</b>	Código	FODO-42
		Versión	01
		Página	1/1
		Sección ISO	
		Fecha de emisión	25 de junio de 2014
Secretaría/División: División de Ingeniería Eléctrica		Área/Departamento: Laboratorios de computación salas A y B	

# Laboratorios de computación salas A y B

*Profesor:* CRUZ NAVARRO JESUS M.C

*Asignatura:* Modelos de programación orientada a objetos

*Grupo:* 1

*No de Práctica(s):* 4

*Integrante(s):* Cortés García Jaime Rafael

*Semestre:* 2017 - 2

*Fecha de entrega:* 09 de Marzo del 2017

*Observaciones:*

**CALIFICACIÓN:**

## Ejercicio :

**1.** Diseñe clase ListaOrdenada, que represente una lista ligada (doble o simple), con las siguientes características:

a. Nodos que guarden un entero.

b. Métodos:

a. void Insertar(int valor)

Se inserta un elemento en el lugar que le corresponde de la lista (de menor a mayor), usando el valor. Es decir, la lista siempre debe estar ordenada. Una lista vacía y una lista de un sólo nodo están ordenadas. No se pueden agregar elementos repetidos ni negativos. Si se da el caso, no agregarlo y mostrar un mensaje en consola.

b. int Leer(int i)

Regresa el valor del elemento en la posición i de la lista (comienza en 0). Si no existe elemento en dicha posición, regresar un -1 e imprimir en consola.

c. int Quitar()

Se quita el elemento con mayor valor de la lista.

d. void Imprimir()

Se imprimen los valores de los nodos desde el primero a hasta el último de la lista, en un solo renglón y separados por comas.

e. int Longitud( )

Regresa el número de elementos de la lista.

**2.** Desarrolle un programa que muestre el funcionamiento de todos los métodos de las clases.

a. Crear una lista ligada.

b. Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.

c. Quitar un elemento.

d. Imprimir los elementos en las posiciones 0, 2 y -1 y el último elemento.

e. Imprimir la longitud de la lista.

f. Agregar los elementos 2 y 4 e imprimir la lista.

## MENU:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default.  _
```

## INCISO B:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default.  B

----ERROR----  [-4]    Número negativo!!

----ERROR----  [5]     Número repetido!!

Lista:  [3, 5, 6, 7, 9]
```

## INCISO C:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default. C

Lista: [3, 5, 6, 7]
```

## INCISO D:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default. D

Lista: [3, 5, 6, 7]

----ERROR---- [0] Número no encontrado!!
----ERROR---- [2] Número no encontrado!!
----ERROR---- [-1] Número no encontrado!!

[7] Se encuentra en el índice: 3
```

## INCISO E:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default.  E

Lista:  [3, 5, 6, 7]

La longitud es de : 4
```

## INCISO F:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default.  F

Lista:  [2, 3, 4, 5, 6, 7]
```

## INCISO G:

```
file:///C:/Users/Raffa/documents/visual studio 2015/Projects/Practica_4/Practica_4/bin/Debug/Practica_4.EXE

MENU

Elige el inciso que desees:

B -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
C -- Quitar un elemento.
D -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
E -- Imprimir la longitud de la lista
F -- Agregar los elementos 2 y 4 e imprimir la lista.
G -- Salir.

La Lista ya ha sido creada por default.  G

Bye bye, Adios!!
_
```

### 3. EXTRA (10 puntos): Implemente el mismo programa en C y compare los códigos.

Para crear el mismo programa en C, empecé copiando absolutamente el mismo código que cree en C# a C, y poco a poco fui modificando todos los "Console.WriteLine( )" por "p", letra a cual, al principio del código en C, la define como "printf", para ahorrarme un poco de trabajo, también cambie los "Console.ReadLine( )" por "s", que de igual manera, al principio del código, define como "scanf".

El código que contenía la clase "Nodo" no lo copie como tal, sino que implemente mi estructura en C que tuviera los mismos componentes.

Otra cosa que tuve que cambiar fue las opciones para el MENU, en C# no tuve problema para que la opción fuera un carácter, sin embargo, al querer compilar el mismo código en C, salía algún error, así que no me complique la existencia, y los cambie por números.

Otro problema con el que me tope, fue en las líneas en donde uso una función en C que regresa un bool, no estaba seguro si en C no funcionaban los bool como en C#, así que opte por cambiar la función para que regresara un int y use el conocimiento de que, en C, al número "1", el programa lo reconoce como verdadero y al "0" como falso. Así que en las partes de bool, en vez de inicializarlas como true, creaba un int que inicializaba como "1". Y modificando así toda esa función, el error desapareció.

Otra de las cosas que tuve que cambiar en todas las funciones en C, fue pasarles tanto head como tail, para que la función supiera donde estaban esas partes de la Lista, y así poder trabajar con ella, insertando, quitando nodos y recorriéndola. A diferencia de C#, al que no se le pasaban ni head, ni tail, lo cual hace notar de que, en C#, puedes acceder muy fácil a diferentes cosas en todos los pedazos de código, si tienes el permiso adecuado.

Sería también importante resaltar que, en C, solo use dos librerías, la clásica <stdio.h>, y para poder jugar con la memoria dinámica, también tuve que usar <stdlib.h>, y por lo cual, una diferencia notable entre los dos códigos, es que en C, se usa malloc para asignar el pedazo de memoria que utilizara el nuevo nodo, y en C#, lo que se creaba era un objeto de la Clase Nodo. En C, ahora hablando de la eliminación de nodos, existe la función "free" que libera ese espacio de memoria, mientras que en C# solo hace falta perder la referencia para considerar a un nodo, eliminado, después vendrá garbage collector para vaciar esa memoria que se desperdicia.

Un punto también que note que no se si pueda ocurrir en C#, es: ¿que pasaria con el programa si no se puede crear una clase de tipo Nodo?, en C, si existía esta consideración, algo improbable para los fines de estos ejercicios, pero me acostumbre en C, que, después de usar malloc para asignar memoria, si no había suficiente memoria, había que validar con un if, si se había hecho esa asignación con éxito, de lo contrario devolvería NULL y si se seguía corriendo el código, las operaciones siguientes, no tendrían sentido para el programa y lanzaría un error, en C, implemente ese pedazo de código, y repito, no sé si en C# pase algo parecido.

Finalmente la última cosa que tuve que modificar del código original en C#, fue simplemente la forma en que se accede a las variables del nodo, en C#, se iban colocando puntos para acceder a las componentes del nodo, mientras que en C, se acostumbra a usar más este símbolo: "->".

## MENU:

```
C:\Users\Raffa\Desktop\Practica_4_Comparaci³4n.exe

MENU

Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default. _
```

## INCISO 2:

```
C:\Users\Raffa\Desktop\Practica_4_Comparaci³4n.exe

MENU

Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default. 2

----ERROR---- [-4]    Numero negativo!!
----ERROR---- [5]     Numero repetido!!

Lista: [3, 5, 6, 7, 9]_
```



### INCISO 3:

```
C:\Users\Raffa\Desktop\Practica_4_Comparaci3\n.exe

MENU

Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default. 3

Lista: [3, 5, 6, 7]
```

### INCISO 4:

```
C:\Users\Raffa\Desktop\Practica_4_Comparaci3\n.exe

MENU

Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default. 4

Lista: [3, 5, 6, 7]

----ERROR---- [0] N-mero no encontrado!!
----ERROR---- [2] N-mero no encontrado!!
----ERROR---- [-1] N-mero no encontrado!!
[7] Se encuentra en el indice: 3
```

## INCISO 5:

```
C:\Users\Raffa\Desktop\Practica_4_Comparaci34n.exe

MENU

Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default.  5

Lista:  [3, 5, 6, 7]

La longitud es de : 4
```

## INCISO 6:

```
C:\Users\Raffa\Desktop\Practica_4_Comparaci34n.exe

MENU

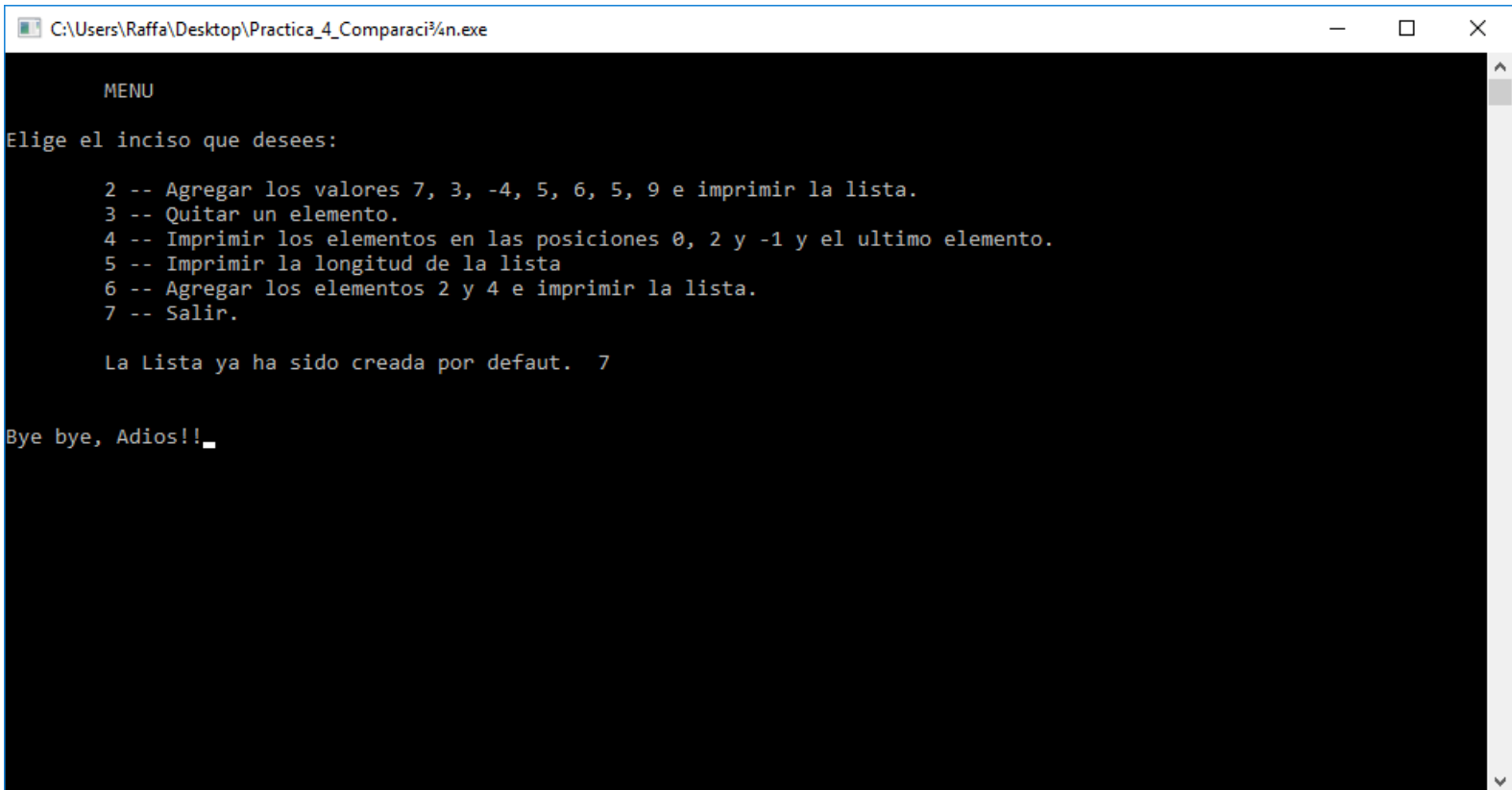
Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default.  6

Lista:  [2, 3, 4, 5, 6, 7]
```

## INCISO 7:



```
C:\Users\Raffa\Desktop\Practica_4_Comparaci%4n.exe

MENU

Elige el inciso que desees:

2 -- Agregar los valores 7, 3, -4, 5, 6, 5, 9 e imprimir la lista.
3 -- Quitar un elemento.
4 -- Imprimir los elementos en las posiciones 0, 2 y -1 y el ultimo elemento.
5 -- Imprimir la longitud de la lista
6 -- Agregar los elementos 2 y 4 e imprimir la lista.
7 -- Salir.

La Lista ya ha sido creada por default. 7

Bye bye, Adios!!_
```

## Conclusión:

Creo que debo estar agradecido por haber entendido bien el concepto de pilas, colas y listas en Estructura de Datos y Algoritmos, ya que ahora en C# al querer implementar ese concepto, no fue realmente algo tan complicado, pero reconozco que este tema no es un tema fácil, ya que puede resultar a veces muy confuso, debo admitir que, en su momento, si me dieron mucha lata al querer implementarlas en C, pero al final, comprendí su funcionamiento y estructura.

Evidentemente estos conceptos se facilitan aquí en C#, y no trabajar con apuntadores suaviza demasiado la abstracción que se debe realizar para poder implementar el código. Tal como dijo el profesor, decidí hacer una Lista Doble porque te da mas control sobre la misma lista, requiere un poco más de coco, pero no demasiado si entiendes la esencia de estas estructuras dinámicas y además se vuelve una ventaja con las operaciones que desees realizar con tu Lista.