



MATERIA
ESTRUCTURA DE DATOS I

“Tema 7. Listas”

Docente: Ing. Ubaldo Pérez Ferreira

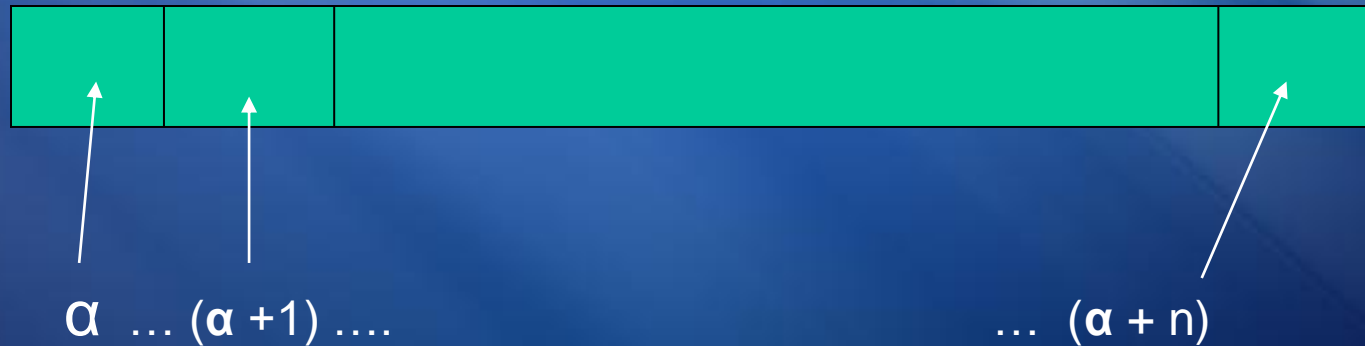
E-mail: ubaperez@gmail.com

Santa Cruz de la Sierra – Bolivia ©
2009 - 2021



```
<?xml version="1.0" encoding="UTF-8" ?>
<wml>
  <card title="TV Interface">
    <channel="channel1.wml"> Bienvenidos a Wireless Tuna </a>
    <a href="#"> canal.wml </a> con el balde de 2003 </a>
    <a href="#"> back.wml </a> back </a>
  </card>
</wml>
```

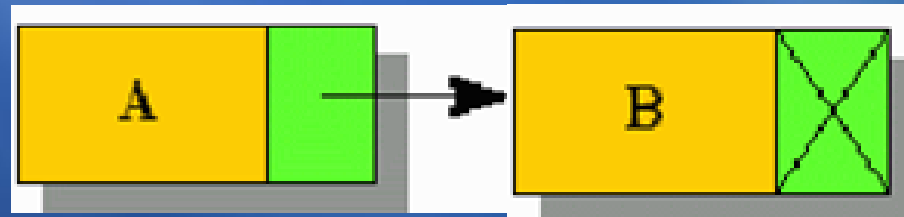
Hasta ahora solamente hemos implementados TAD utilizando “**Estructura de Datos Estática**”, y es aquella en la que el tamaño ocupado en memoria se define antes de que el programa se ejecute y no pueda modificarse dicho tamaño durante la ejecución del programa, entre las estructuras de datos estáticas se encuentran los array (vectores y matrices), registros, archivos y cadenas.



```
<?xml version="1.0" encoding="UTF-8" ?>
<wml>
  <card title="TV Interface">
    <chref="channel.wml"> Bit Cerco Wireless Tune</a>
    <a href="ani.wml"> ani con o bal des: 2003</a>
    <a href="back.wml"> back</a>
  </card>
</wml>
```

Estructura Estáticas vs Dinámica

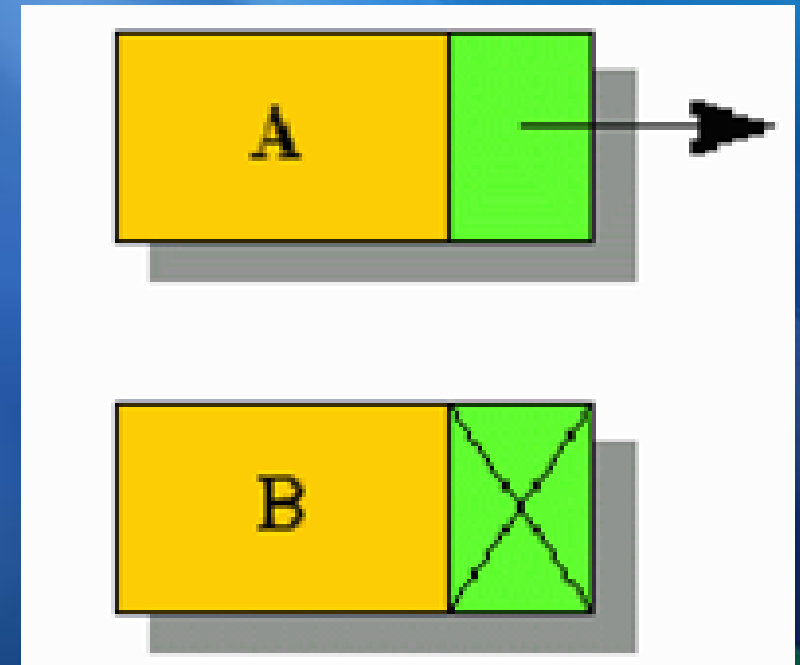
La Estructura de Datos Dinámica, no tienen las limitaciones o restricciones en el tamaño de memoria ocupada. Mediante el uso de un tipo de datos específico, denominado **puntero**, es posible construir estructuras de datos dinámicas, estas características ofrecen soluciones eficaces y efectivas en la solución de problemas complejos.



```
<?xml version="1.0" encoding="UTF-8" ?>
<wml>
  <card title="TV Interface">
    <chref="channel1.wml"> Bit Cerco Wireless Tune </a>
    <a href="ani.wml"> ani con o bal des: 2003 </a>
    <a href="back.wml"> back </a>
  </card>
</wml>
```

Un nodo es una estructura o registro que dispone de varios campos y al menos uno de esos campos es puntero o referencia a otro nodo.

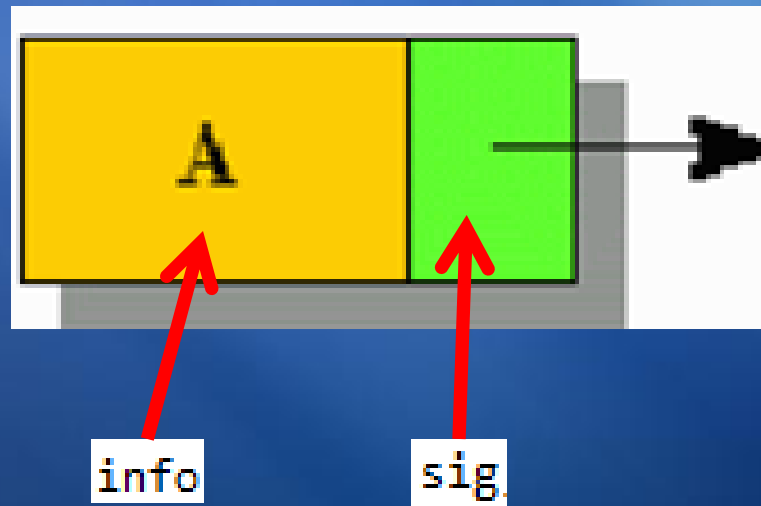
- Cada nodo se divide en áreas de contenido (en naranja) y una o más áreas de enlace (en verde).
- La área de enlace de A tiene una flecha para indicar que referencian a otro nodo del mismo tipo (o subtipo).
- La única área de enlace de B incorpora una X (Null) para indicar una referencia nula. En otras palabras, B no está conectado a ningún otro nodo

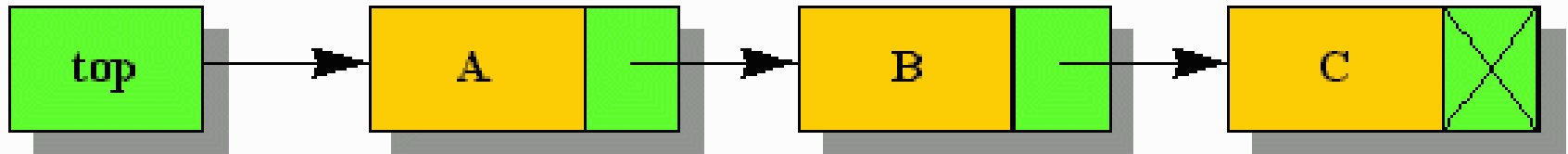


```
<wml>
<card title="TV Interface">
  <channel="channel1.wml"> Big Car: Wireless Tune </a>
  <a href="uni.wml"> uni con o bal des: 2003 </a>
  <a href="back.wml"> back </a>
</card>
</wml>
```

Implementación de un Nodo

```
// Se define la clase Nodo  
class Nodo  
{  
    public int info;  
    public Nodo sig;  
}
```



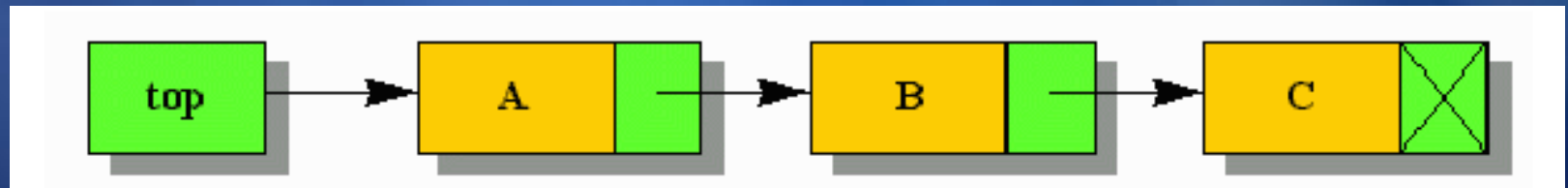


- Son estructuras de datos secuenciales de cero o más elementos (nodo) de un tipo dado almacenados en memoria.
- Cada elemento de la lista, excepto el primero, tiene un único predecesor y cada elemento de la lista, excepto el último, tiene un único sucesor.
- El número de elementos de la lista se llama longitud. Si tiene 0 elementos se llama **lista vacía**.
- En una lista podemos añadir nuevos elementos o suprimirlos en cualquier posición.



```
<wml>
<card title="TV Interface">
  <channel="channel1.wml"> Big Cero: Wireless Time </a>
  <a href="ani.wml"> ani con o bal des: 2003 </a>
  <a href="back.wml"> back </a>
</card>
</wml>
```

- Una **lista de enlace simple**, es una lista enlazada de nodos, donde cada nodo tiene un único campo de enlace.
- Una variable de referencia contiene una referencia al primer nodo, cada nodo (excepto el último) enlaza con el nodo siguiente, y el enlace del último nodo contiene null para indicar el final de la lista.

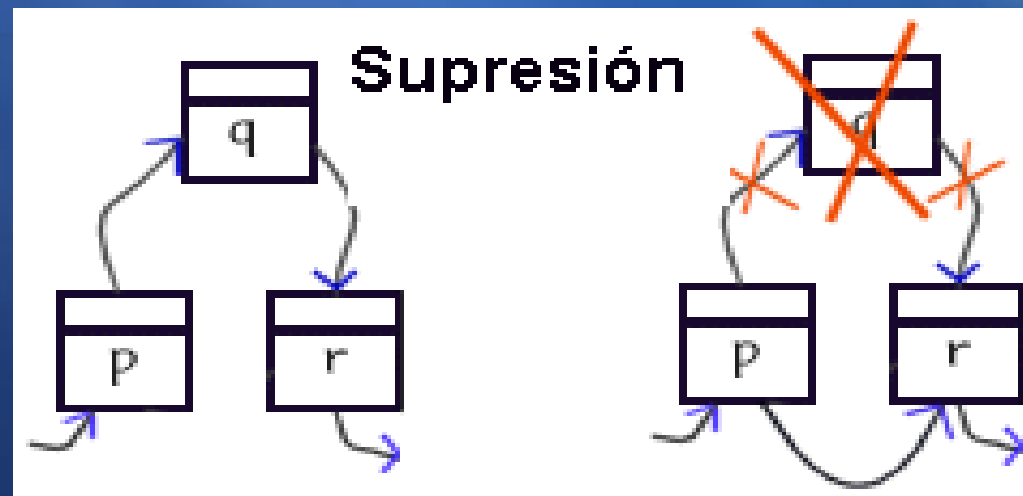
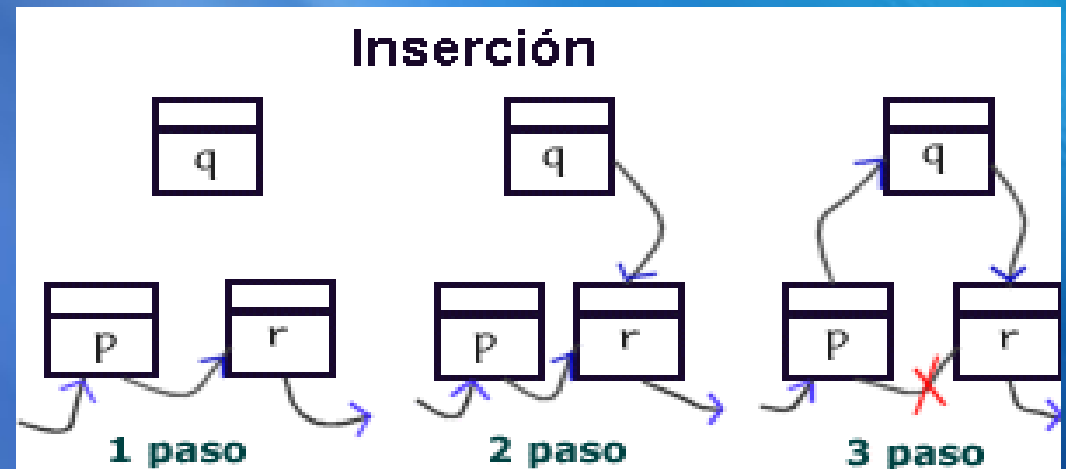


```
<?xml version="1.0" encoding="UTF-8" ?>
<wml>
  <card title="TV Interface">
    <chref="channel.wml"> Bit Cerco Wireless Tune</ch>
    <ca href="ami.wml"> ami con o bal des: 2003</ca>
    <ca href="back.wml"> back</ca>
  </card>
</wml>
```

Operaciones sobre Listas Enlazadas

Algunas operaciones son:

- Inserción
- Supresión
- Recorrido
- Ordenación
- Búsqueda
- Consulta



```
<wml>  
<card title="TV Interface">  
  <channel="channel1.wml"> Bit Cerco Wireless Tune</a>  
  <a href="ani.wml"> ani con o bal des 2003</a>  
  <a href="back.wml"> back</a>  
</card>  
</wml>
```


Structure Lista

- | | |
|------------------------------------|--|
| 1. Define Create() → Lista | //Crea la Lista |
| 2. Vacio(Lista)→ Boolean | // Esta vacia |
| 3. Primero(Lista)→nodo | // Primer nodo de la Lista (Tope) |
| 4. Ultimo(Lista)→nodo | // Ultimo Nodo de La Lista (Rear) |
| 5. Siguierte(nodo)→nodo | // Sucesor de un Nodo |
| 6. Anterior(nodo)→nodo | // Antecesor de un Nodo |
| 7. Buscar(Lista, valor)→nodo | // Busca un Valor en la Lista |
| 8. Longitud(Lista)→int | // Tamaño de la lista |
| 9. AddA(Lista, valor,nodo)→Lista | // Inserta un valor en la Lista antes de |
| 10.AddD(Lista, valor, nodo) →Lista | // Inserta un valor en la Lista después de |
| 11.Delete(Lista, valor)→Lista | // Borra un nodo con un Valor de la Lista |



```
<?xml version="1.0" encoding="UTF-8" ?>
<card title="TV Interface">
  <channel="channel1.xml"> Bit Cerco Wireless Tune </a>
  <a href="ami.xml"> ami con o bal des 2003 </a>
  <a href="back.xml"> back </a>
</card>
</xml>
```

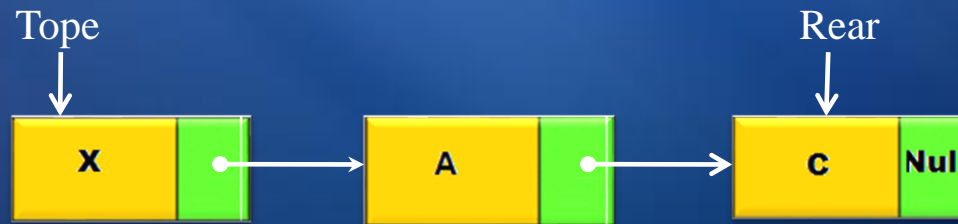

Implementación Listas Enlazadas

```
public class clsLista
{
    // Se define la clase Nodo
    class Nodo
    {
        public int info;
        public Nodo sig;
    }

    // Se define los nodos tope (primero) y rear (ultimo)
    // se puede implementar con un solo puntero pero
    // cada vez tenemos que recorrer toda la lista para encontrar el ultimo

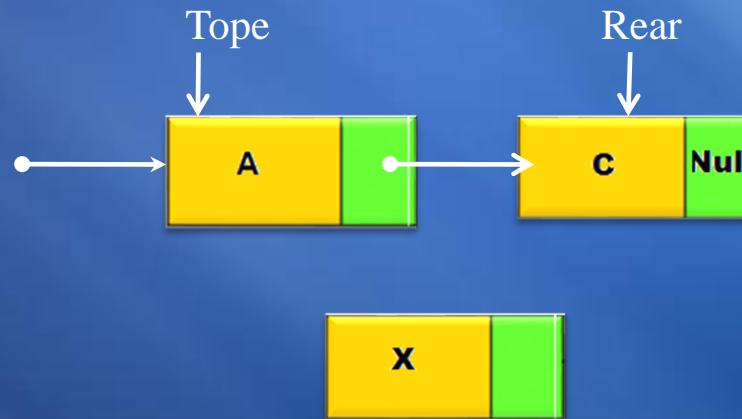
    private Nodo tope, rear;

    // Constructor de la Clase Lista
    public clsLista()
    {
        tope = null;
        rear = null;
    }
}
```



Inserción de un nodo al inicio de la Listas

El nuevo nodo X se coloca al principio de la lista, convirtiéndose en el primero de la misma.



```
// Adiciona un Nodo al inicio de la Lista
public void AddPrimero(int info)
{
    Nodo nuevo;
    nuevo = new Nodo();
    nuevo.info = info;
    nuevo.sig = tope;

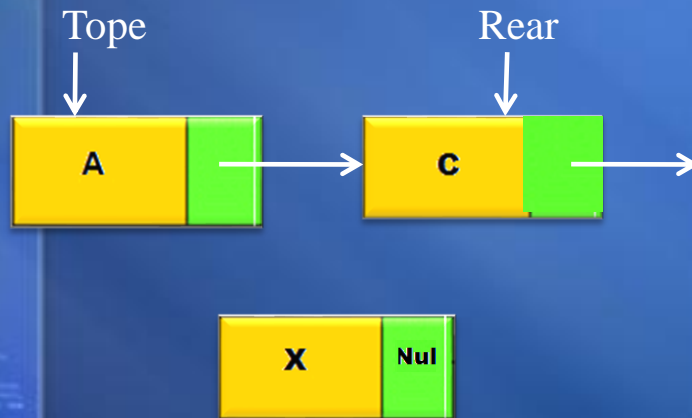
    if (Vacia())
    {
        tope = nuevo;
        rear = nuevo;
    }
    else
    {
        tope = nuevo;
    }
}
```



```
<wml>
<card title="TV Interface">
  <channel="channel1.wml"> Bit Cerco Wireless Tune</a>
  <a href="ani.wml"> ani con o bal des 2003</a>
  <a href="back.wml"> back</a>
</card>
</wml>
```


Inserción de un nodo al final de la Listas

El nuevo nodo se coloca al final de la lista, convirtiéndose en el ultimo de la misma.



```
//Adiciona una Nodo al Final de la Lista
public void AddUltimo(int info)
{
    Nodo nuevo;
    nuevo = new Nodo();
    nuevo.info = info;
    nuevo.sig = null;
    if (Vacia())
    {
        tope = nuevo;
        rear = nuevo;
    }
    else
    {
        rear.sig = nuevo;
        rear = nuevo;
    }
}
```



```
</wml>
<card title="TV Interface">
  <channel="channel1.wml"> Bit Cerco Wireless Tune </a>
  <a href="ani.wml"> ani con o bal des: 2003 </a>
  <a href="back.wml"> back </a>
</card>
</wml>
```

Inserción de un nodo antes de otro

El nuevo nodo se coloca antes/después de otro nodo dado como referencia. Primero se tiene que realizar la búsqueda del nodo de cómo referencia.



```
// Adciona un Nodo Antes de un Nodo X dado
public bool AddAntes( int info,int infoX)
{
    // Buscamos el Nodo X en la Lista
    // recorremos toda la lista desde el Tope hasta el Rear

    Nodo anterior = null;    // El anterior parte en Nulo
    Nodo actual = tope;      // El Actual es igual al Primero

    while (actual != null)
    {
        if (actual.info == infoX)    // Encontro el nodo y se inserta el nuevo antes de
        {
            Nodo nuevo;
            nuevo = new Nodo();
            nuevo.info = info;
            nuevo.sig = actual;    // El nuevo Nodo apunta al actual Nodo
            anterior.sig = nuevo;  // El anterior Nodo apunta al Nuevo nodo
            return true;
        }
        anterior = actual;    // Anterior es igual al nodo actual
        actual = actual.sig;   // Recorremos el Siguiente Nodo
    }
    return false;
}
```



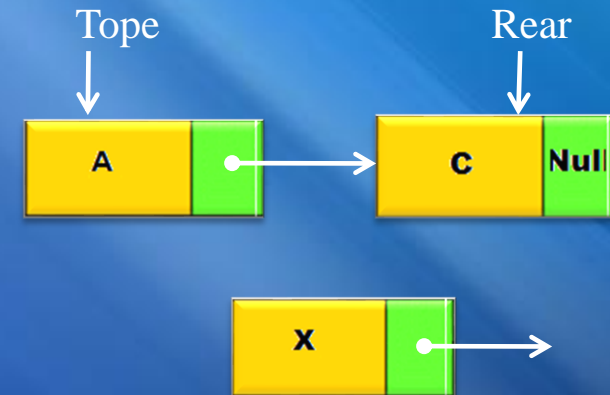
Inserción de un nodo después de otro

El nuevo nodo se coloca antes/después de otro nodo dado como referencia. Primero se tiene que realizar la búsqueda del nodo de cómo referencia.

```
// Adciona Un Nodo Despues de un Nodo X dado
public bool AddDespues(int info, int infoX)
{
    // Buscamos el Nodo X en la Lista
    // recorremos toda la lista desde el Tope hasta el Rear

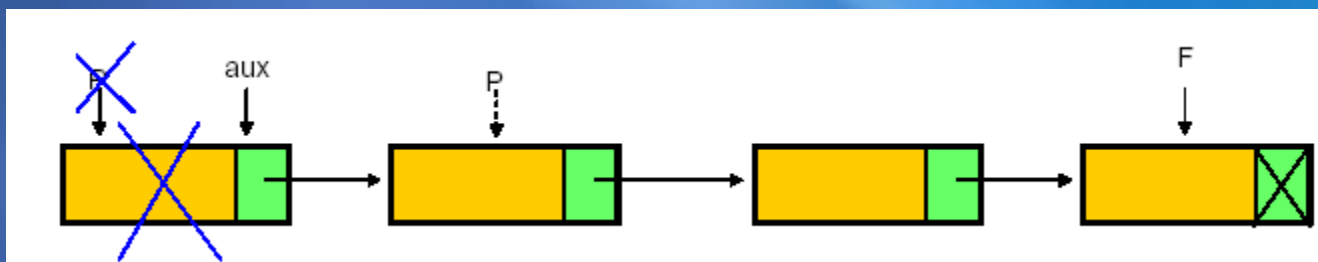
    Nodo siguiente = tope;    // El siguienete Nodo es igual al tope
    Nodo actual = tope;      // El nodo actual es igual al Tope
    siguiente = siguiente.sig; // Recorremos el siguiente un nodo mas
    while (actual != null)
    {
        if (actual.info == infoX)
        {
            Nodo nuevo;
            nuevo = new Nodo();
            nuevo.info = info;
            nuevo.sig = siguiente; // El nuevo Nodo apunta al siguiente
            actual.sig = nuevo;    // El actual Nodo apunta al nuevo

            return true;
        }
        actual = actual.sig;
        siguiente = siguiente.sig;
    }
    return false;
}
```

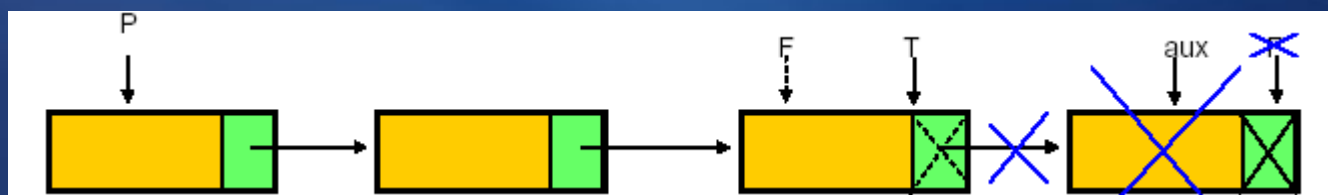


Eliminación de un elemento o nodo

Eliminación primer nodo: se quita el primer elemento de la lista, redefiniendo el valor del puntero al nuevo inicio de la lista.



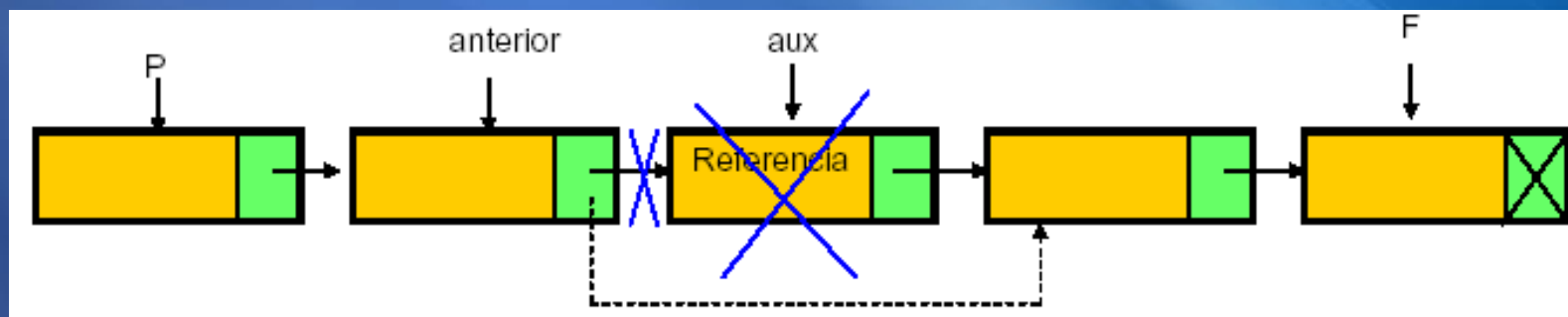
Eliminación del último nodo: se quita el último nodo de la lista, redefiniendo a null el campo liga o enlace del penúltimo nodo de la lista. Para encontrar el penúltimo nodo es necesario recorrer la lista. Se redefine el valor del puntero que indica el fin de la lista.



```
<wml>
<card title="TV Interface">
  <channel="channel1.wml"> Bit Cerco Wireless Tune</a>
  <a href="ani.wml"> ani con o bal des 2003</a>
  <a href="back.wml"> back</a>
</card>
</wml>
```

Eliminación de un elemento o nodo

- **Eliminación un nodo con información X:** se debe buscar al nodo que contenga la información dada como referencia (X) y eliminarlo, estableciendo la liga o enlace correspondiente entre su predecesor y su sucesor.



```
</wml>
<card title="TV Interface">
  <channel="channel1.wml"> Bit Cerco Wireless Tune </a>
  <a href="ani.wml"> ani con o bal des 2003 </a>
  <a href="back.wml"> back </a>
</card>
</wml>
```


Eliminación de un elemento o nodo

```
// Borrar un nodo dado X
public void Delete(int info)
{
    Nodo anterior, nodo;
    nodo = tope;
    anterior = null;
    Boolean existe = false;
    while (nodo != null && existe==false)    // recorremos la lista hasta encontrar el nodo
    {
        if (nodo.info == info)
            existe = true;
        else
        {
            anterior = nodo;
            nodo = nodo.sig;
        }
    }

    if (existe == false )    // no se encuentra el nodo
        return;

    if (nodo == tope)    // si el nodo es el primero
    {
        tope = nodo.sig;    // se actualiza el puntero que apunta al primero
        nodo = null;
        return;
    }

    if (nodo == rear)    // si el nodo es el ultimo
    {
        anterior.sig = null;
        rear = anterior;    // se actualiza el puntero que apunta al ultimo
        nodo = null;
        return;
    }

    anterior.sig = nodo.sig;    // el nodo esta en medio de dos nodos
}
```



- La operación de recorrido consiste en visitar cada uno de los nodos que forman la lista.
- La visita de un nodo puede definirse por medio de una operación muy simple (como escribir su valor), o por medio de operaciones tan complejas como se desee.
- Para recorrer todos los nodos de una lista se comienza con el primero. Tomando el valor del campo enlace o liga se avanza por cada nodo hasta que dicho enlace sea igual a NULL.

```
// Recorre la Lista
public string View()
{
    string cad = "";
    Nodo reco = tope;
    while (reco != null)
    {
        cad+=reco.info + "->";
        reco = reco.sig;
    }
    return cad;
}
```



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Channel 11 -->
<channel name="Channel 11" url="http://www.11.com" />
<!-- Channel 12 -->
<channel name="Channel 12" url="http://www.12.com" />
<!-- Channel 13 -->
<channel name="Channel 13" url="http://www.13.com" />
<!-- Channel 14 -->
<channel name="Channel 14" url="http://www.14.com" />
<!-- Channel 15 -->
<channel name="Channel 15" url="http://www.15.com" />
<!-- Channel 16 -->
<channel name="Channel 16" url="http://www.16.com" />
<!-- Channel 17 -->
<channel name="Channel 17" url="http://www.17.com" />
<!-- Channel 18 -->
<channel name="Channel 18" url="http://www.18.com" />
<!-- Channel 19 -->
<channel name="Channel 19" url="http://www.19.com" />
<!-- Channel 20 -->
<channel name="Channel 20" url="http://www.20.com" />
<!-- Channel 21 -->
<channel name="Channel 21" url="http://www.21.com" />
<!-- Channel 22 -->
<channel name="Channel 22" url="http://www.22.com" />
<!-- Channel 23 -->
<channel name="Channel 23" url="http://www.23.com" />
<!-- Channel 24 -->
<channel name="Channel 24" url="http://www.24.com" />
<!-- Channel 25 -->
<channel name="Channel 25" url="http://www.25.com" />
<!-- Channel 26 -->
<channel name="Channel 26" url="http://www.26.com" />
<!-- Channel 27 -->
<channel name="Channel 27" url="http://www.27.com" />
<!-- Channel 28 -->
<channel name="Channel 28" url="http://www.28.com" />
<!-- Channel 29 -->
<channel name="Channel 29" url="http://www.29.com" />
<!-- Channel 30 -->
<channel name="Channel 30" url="http://www.30.com" />
<!-- Channel 31 -->
<channel name="Channel 31" url="http://www.31.com" />
<!-- Channel 32 -->
<channel name="Channel 32" url="http://www.32.com" />
<!-- Channel 33 -->
<channel name="Channel 33" url="http://www.33.com" />
<!-- Channel 34 -->
<channel name="Channel 34" url="http://www.34.com" />
<!-- Channel 35 -->
<channel name="Channel 35" url="http://www.35.com" />
<!-- Channel 36 -->
<channel name="Channel 36" url="http://www.36.com" />
<!-- Channel 37 -->
<channel name="Channel 37" url="http://www.37.com" />
<!-- Channel 38 -->
<channel name="Channel 38" url="http://www.38.com" />
<!-- Channel 39 -->
<channel name="Channel 39" url="http://www.39.com" />
<!-- Channel 40 -->
<channel name="Channel 40" url="http://www.40.com" />
<!-- Channel 41 -->
<channel name="Channel 41" url="http://www.41.com" />
<!-- Channel 42 -->
<channel name="Channel 42" url="http://www.42.com" />
<!-- Channel 43 -->
<channel name="Channel 43" url="http://www.43.com" />
<!-- Channel 44 -->
<channel name="Channel 44" url="http://www.44.com" />
<!-- Channel 45 -->
<channel name="Channel 45" url="http://www.45.com" />
<!-- Channel 46 -->
<channel name="Channel 46" url="http://www.46.com" />
<!-- Channel 47 -->
<channel name="Channel 47" url="http://www.47.com" />
<!-- Channel 48 -->
<channel name="Channel 48" url="http://www.48.com" />
<!-- Channel 49 -->
<channel name="Channel 49" url="http://www.49.com" />
<!-- Channel 50 -->
<channel name="Channel 50" url="http://www.50.com" />
<!-- Channel 51 -->
<channel name="Channel 51" url="http://www.51.com" />
<!-- Channel 52 -->
<channel name="Channel 52" url="http://www.52.com" />
<!-- Channel 53 -->
<channel name="Channel 53" url="http://www.53.com" />
<!-- Channel 54 -->
<channel name="Channel 54" url="http://www.54.com" />
<!-- Channel 55 -->
<channel name="Channel 55" url="http://www.55.com" />
<!-- Channel 56 -->
<channel name="Channel 56" url="http://www.56.com" />
<!-- Channel 57 -->
<channel name="Channel 57" url="http://www.57.com" />
<!-- Channel 58 -->
<channel name="Channel 58" url="http://www.58.com" />
<!-- Channel 59 -->
<channel name="Channel 59" url="http://www.59.com" />
<!-- Channel 60 -->
<channel name="Channel 60" url="http://www.60.com" />
<!-- Channel 61 -->
<channel name="Channel 61" url="http://www.61.com" />
<!-- Channel 62 -->
<channel name="Channel 62" url="http://www.62.com" />
<!-- Channel 63 -->
<channel name="Channel 63" url="http://www.63.com" />
<!-- Channel 64 -->
<channel name="Channel 64" url="http://www.64.com" />
<!-- Channel 65 -->
<channel name="Channel 65" url="http://www.65.com" />
<!-- Channel 66 -->
<channel name="Channel 66" url="http://www.66.com" />
<!-- Channel 67 -->
<channel name="Channel 67" url="http://www.67.com" />
<!-- Channel 68 -->
<channel name="Channel 68" url="http://www.68.com" />
<!-- Channel 69 -->
<channel name="Channel 69" url="http://www.69.com" />
<!-- Channel 70 -->
<channel name="Channel 70" url="http://www.70.com" />
<!-- Channel 71 -->
<channel name="Channel 71" url="http://www.71.com" />
<!-- Channel 72 -->
<channel name="Channel 72" url="http://www.72.com" />
<!-- Channel 73 -->
<channel name="Channel 73" url="http://www.73.com" />
<!-- Channel 74 -->
<channel name="Channel 74" url="http://www.74.com" />
<!-- Channel 75 -->
<channel name="Channel 75" url="http://www.75.com" />
<!-- Channel 76 -->
<channel name="Channel 76" url="http://www.76.com" />
<!-- Channel 77 -->
<channel name="Channel 77" url="http://www.77.com" />
<!-- Channel 78 -->
<channel name="Channel 78" url="http://www.78.com" />
<!-- Channel 79 -->
<channel name="Channel 79" url="http://www.79.com" />
<!-- Channel 80 -->
<channel name="Channel 80" url="http://www.80.com" />
<!-- Channel 81 -->
<channel name="Channel 81" url="http://www.81.com" />
<!-- Channel 82 -->
<channel name="Channel 82" url="http://www.82.com" />
<!-- Channel 83 -->
<channel name="Channel 83" url="http://www.83.com" />
<!-- Channel 84 -->
<channel name="Channel 84" url="http://www.84.com" />
<!-- Channel 85 -->
<channel name="Channel 85" url="http://www.85.com" />
<!-- Channel 86 -->
<channel name="Channel 86" url="http://www.86.com" />
<!-- Channel 87 -->
<channel name="Channel 87" url="http://www.87.com" />
<!-- Channel 88 -->
<channel name="Channel 88" url="http://www.88.com" />
<!-- Channel 89 -->
<channel name="Channel 89" url="http://www.89.com" />
<!-- Channel 90 -->
<channel name="Channel 90" url="http://www.90.com" />
<!-- Channel 91 -->
<channel name="Channel 91" url="http://www.91.com" />
<!-- Channel 92 -->
<channel name="Channel 92" url="http://www.92.com" />
<!-- Channel 93 -->
<channel name="Channel 93" url="http://www.93.com" />
<!-- Channel 94 -->
<channel name="Channel 94" url="http://www.94.com" />
<!-- Channel 95 -->
<channel name="Channel 95" url="http://www.95.com" />
<!-- Channel 96 -->
<channel name="Channel 96" url="http://www.96.com" />
<!-- Channel 97 -->
<channel name="Channel 97" url="http://www.97.com" />
<!-- Channel 98 -->
<channel name="Channel 98" url="http://www.98.com" />
<!-- Channel 99 -->
<channel name="Channel 99" url="http://www.99.com" />
<!-- Channel 100 -->
<channel name="Channel 100" url="http://www.100.com" />
</xml>
```

Búsqueda de un elemento

- La operación de búsqueda de un elemento en lista se realiza de modo secuencial.
- Se deben recorrer los nodos, tomando el campo enlace o liga como referencia al siguiente nodo a visitar.
- La búsqueda puede optimizarse cuando la lista esta ordenada.

```
// Buscar un nodo x dado
public bool Buscar(int info)
{
    Nodo reco = tope;
    while (reco != null)
    {
        if (reco.info==info)
            return true;
        reco = reco.sig;
    }
    return false;
}
```



```
<wml>
<card title="TV Interface">
  <h1ref="channel.wml">Br Cerco Wireless Tune</h1>
  <a href="uni.wml">ami con o bal des 2003</a>
  <a href="back.wml">back</a>
</card>
</wml>
```