



**INSTITUTO POLITÉCNICO NACIONAL**



**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA y  
ELÉCTRICA  
UNIDAD CULHUACAN**

**INGENIERÍA EN COMPUTACIÓN**

**ASIGNATURA: ESTRUCTURA DE DATOS**

Listas Enlazadas

Suarez Vega Edgar Alan

Profesora: Clarissa Janeth González Acatitla

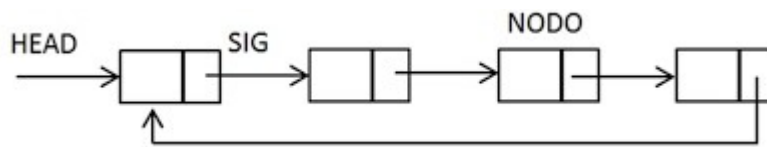
Semestre 2-2021

Febrero - Junio 2021

# Lista Circular

## Funcionamiento

Una lista circularmente enlazada tiene el mismo tipo de nodos que una lista simple enlazada. Esto es, cada nodo en una lista circularmente enlazada tiene un apuntador siguiente y una referencia a un elemento. Pero no hay una cabeza o cola en la lista circularmente enlazada. En vez de tener que el apuntador del ultimo nodo sea null, en una lista circularmente enlazada, este apunta de regreso al primer nodo. Por lo tanto, no hay primer nodo o ultimo. Si se recorren los nodos de una lista circularmente enlazada desde cualquier nodo usando los apuntadores sig, se circulará a través de los nodos.



Es una variación de las listas simples. Consiste en reemplazar el valor NIL o NULL del nexo del último nodo, por la dirección del nodo cabeza de la lista, el cual a su vez contiene la dirección del primer nodo de la lista.

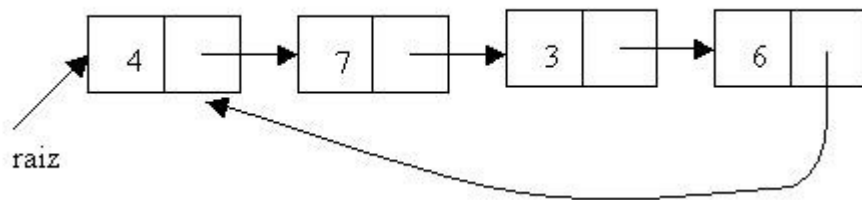
Aun cuando una lista circularmente enlazada no tiene inicio o terminación, no obstante se necesita que algún nodo esté marcado como especial, el cual será llamado el cursor. El nodo cursor permite tener un lugar para iniciar si se requiere recorrer una lista circularmente inversa. Y si se recuerda esta posición inicial, entonces también se puede saber cuando se haya terminado con un recorrido en la lista circularmente enlazada, que es cuando se regresa al nodo que fue el nodo cursor cuando se inicio.

## Operaciones

- **agregar(v):** inserta un nuevo nodo v inmediatamente después del cursor; si la lista está vacía, entonces v se convierte en el cursor y su apuntador sig apunta a el mismo.
- **remover():** borra y regresa el nodo v inmediatamente después del cursor (no el propio cursor, a menos que este sea el único nodo); si la lista queda vacía, el cursor es puesto a null.
- **avanzar():** avanza el cursor al siguiente nodo en la lista.

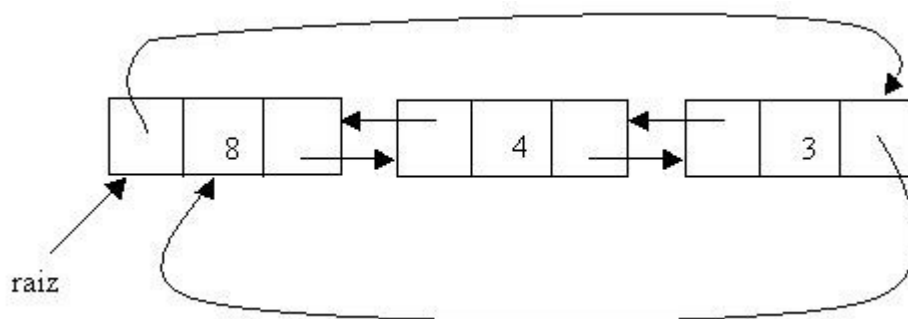
## Ejemplo

Una lista circular simplemente encadenada la podemos representar gráficamente:



Observemos que el puntero sig del último nodo apunta al primer nodo. En este tipo de listas si avanzamos raíz no perdemos la referencia al nodo anterior ya que es un círculo.

Una lista circular puede también ser doblemente encadenada:



El puntero ant del primer nodo apunta al último nodo de la lista y el puntero sig del último nodo de la lista apunta al primero.

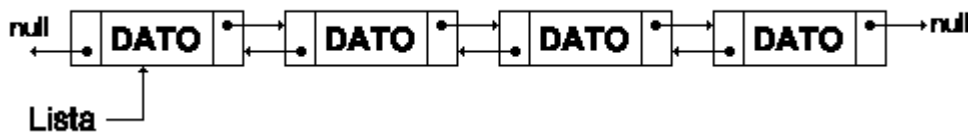
Resolveremos algunos métodos para administrar listas genéricas circulares doblemente encadenadas para analizar la mecánica de enlace de nodos.

# Lista Doblemente Enlazada

## Funcionamiento

Una lista doblemente enlazada es una lista lineal en la que cada nodo tiene dos enlaces, uno al nodo siguiente, y otro al anterior.

Las listas doblemente enlazadas no necesitan un nodo especial para acceder a ellas, pueden recorrerse en ambos sentidos a partir de cualquier nodo, esto es porque a partir de cualquier nodo, siempre es posible alcanzar cualquier nodo de la lista, hasta que se llega a uno de los extremos.



Otra ventaja de las listas doblemente enlazadas es que podemos usar un puntero a la celda que contiene el *i*-ésimo elemento de una lista para representar la posición *i*, mejor que usar el puntero a la celda anterior aunque lógicamente, también es posible la implementación similar a la expuesta en las listas simples haciendo uso de la cabecera. El único precio que pagamos por estas características es la presencia de un puntero adicional en cada celda y consecuentemente procedimientos algo más largos para algunas de las operaciones básicas de listas.

## Operaciones

De nuevo tenemos el mismo repertorio de operaciones sobre este tipo listas:

- Añadir o insertar elementos.
- Buscar o localizar elementos.
- Borrar elementos.
- Moverse a través de la lista, siguiente y anterior.

## Ejemplo

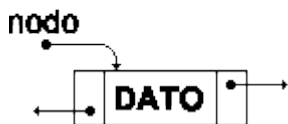
### Añadir un elemento

Nos encontramos ahora ante un tipo de estructura algo diferente de las que hemos estado viendo, así que entraremos en más detalles.

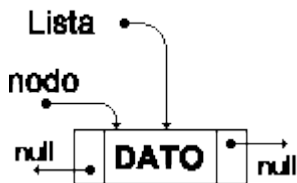
Vamos a intentar ver todos los casos posibles de inserción de elementos en listas doblemente enlazadas.

### Añadir elemento en una lista doblemente enlazada vacía

Lista • → NULL



### Inserción en lista vacía



### Inserción en lista vacía

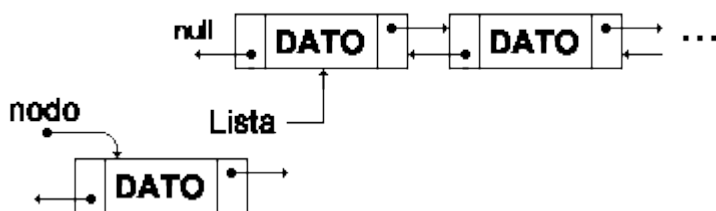
Partiremos de que ya tenemos el nodo a insertar y, por supuesto un puntero que apunte a él, además el puntero que define la lista, que valdrá NULL:

El proceso es muy simple, bastará con que:

1. lista apunta a nodo.
2. lista->siguiente y lista->anterior apunten a null.

### Insertar un elemento en la primera posición de la lista

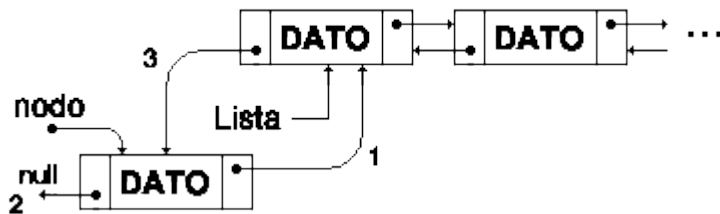
Partimos de una lista no vacía. Para simplificar, consideraremos que lista apunta al primer elemento de la lista doblemente enlazada:



### Insertar en primera posición

El proceso es el siguiente:

1. nodo->siguiente debe apuntar a Lista.
2. nodo->anterior apuntará a Lista->anterior.
3. Lista->anterior debe apuntar a nodo.

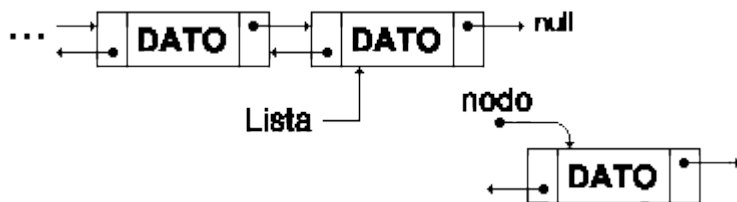


Nodo insertado en primera posición

Recuerda que Lista no tiene por qué apuntar a ningún miembro concreto de una lista doblemente enlazada, cualquier miembro es igualmente válido como referencia.

## Insertar un elemento en la última posición de la lista

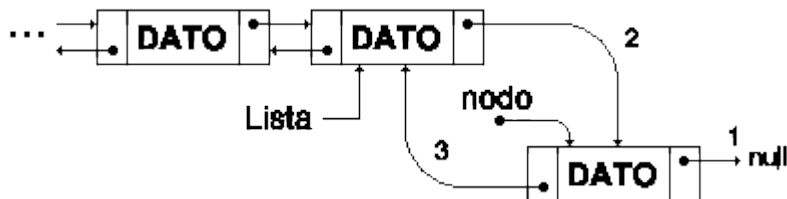
Igual que en el caso anterior, partiremos de una lista no vacía, y de nuevo para simplificar, que Lista está apuntando al último elemento de la lista:



Insertar nodo al final

El proceso es el siguiente:

1. nodo->siguiente debe apuntar a Lista->siguiente (NULL).
2. Lista->siguiente debe apuntar a nodo.
3. nodo->anterior apuntará a Lista.



Nodo insertado en última posición

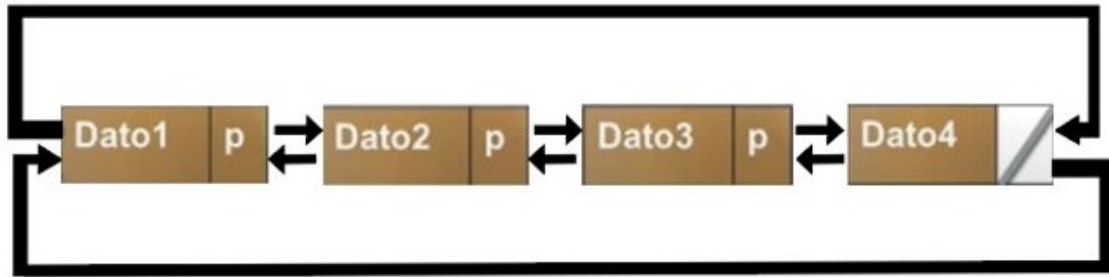
# Circular Doblemente Enlazada

## Funcionamiento

Una lista doblemente enlazada en la que el ultimo elemento se enlaza al primer elemento y viceversa. Esta lista se puede recorrer de modo circular (en anillo) tanto en dirección directa (adelante) como inversa (atrás).

Cada nodo tiene dos enlaces, similares a los de la lista doblemente enlazada, excepto que el enlace anterior del primer nodo apunta al último y el enlace siguiente del último nodo, apunta al primero. Como en una lista doblemente enlazada, las inserciones y eliminaciones pueden ser hechas desde cualquier punto con acceso a algún nodo cercano.

Aunque estructuralmente una lista circular doblemente enlazada no tiene ni principio ni fin, un puntero de acceso externo puede establecer el nodo apuntado que está en la cabeza o al nodo cola, y así mantener el orden tan bien como en una lista doblemente enlazada.

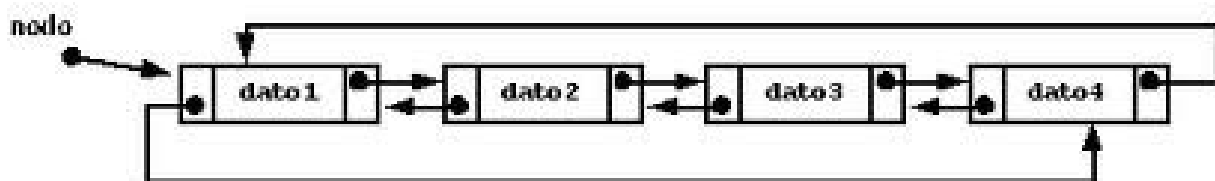


## Operaciones

- Al ser una estructura de datos podemos operar con las operaciones ya conocidas como los es:
- Insertar(Insertar al final, entre dos nodos de la lista, al inicio de)
- Eliminar (Eliminar al final, entre dos nodos de la lista, al inicio de)
- Búsqueda
- Mostrar

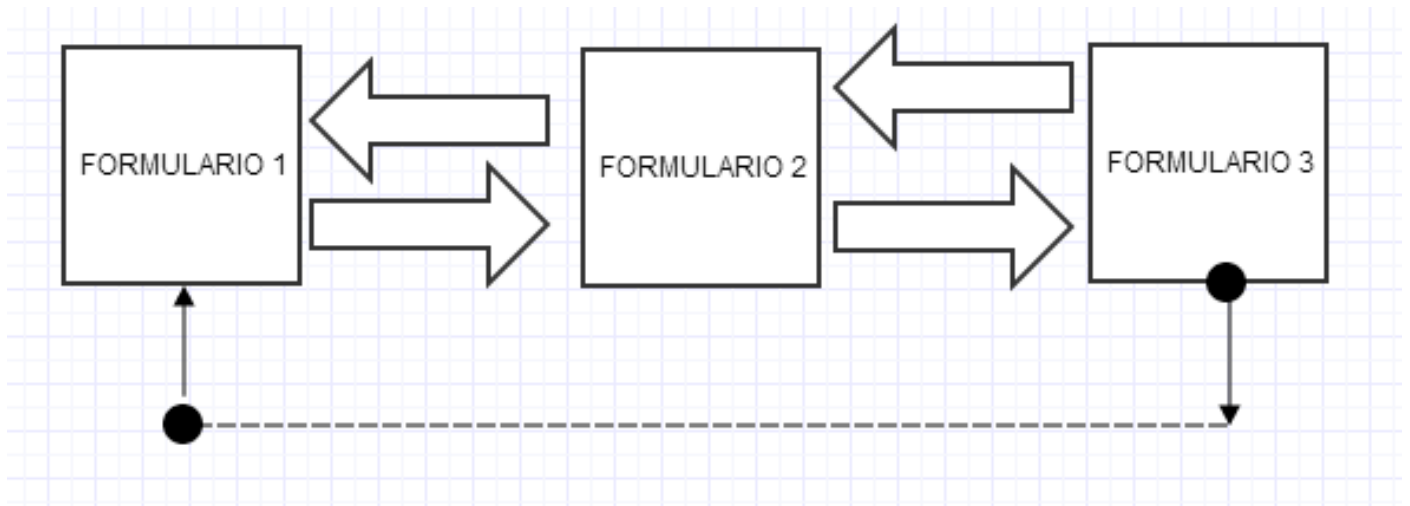
## Ejemplo

Un ejemplo donde se almacenan números y vemos como la estructura donde el último elemento tiene referencia siguiente al primer elemento y la referencia al anterior al primer elemento de la lista también es el último. Y con esto podemos ver que podemos acceder a los dato recorriendo los hacia delante hasta el final o hacia atrás hasta el inicio.





Un formulario electrónico al finalizar cada pagina te da la opción , atrás y siguiente que te permite rectificar cualquier error regresando atrás y adelante y al finalizar comienza nuevamente el formulario para el siguiente usuario



# Referencias

Circular

[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro9/listas\\_circulares.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro9/listas_circulares.html)

<https://www.tutorialesprogramacionya.com/cmasmasya/detalleconcepto.php?codigo=174&punto=43&inicio=30>

Doblemente E

<http://conclase.net/c/edd/cap5>

<https://ccia.ugr.es/~jfv/ed1/tedi/cdrom/docs/ldoble.html>

Circular y Doblemente E

<https://sites.google.com/site/listasenlazadas/lista-enlazadas-doble-circulares>