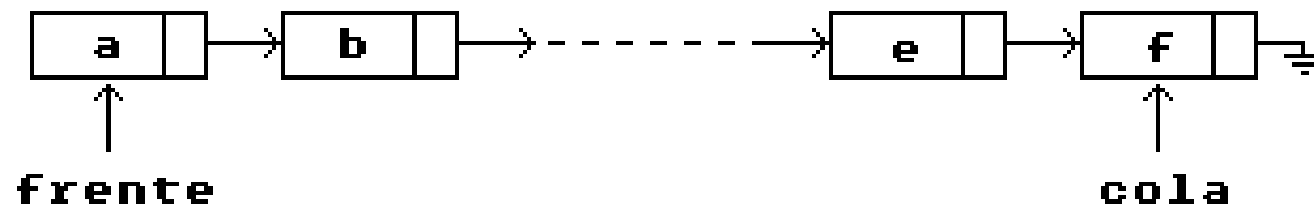


LISTAS LIGADAS

Una lista ligada es una estructura de datos en la que cada elemento (nodo) tiene dos partes, uno con información (un valor de tipo genérico, dato, info, etc.) y el otro con la dirección del nodo siguiente.



Insertión y eliminación de nodos

Una lista es una estructura de datos dinámica.

El número de nodos puede variar dramáticamente.

Para poder crecer la lista necesitamos un mecanismo para generar nodos vacíos:

```
p = getnode()
```

Donde p es un nuevo nodo vacío generado por getnode()

Listas Ligadas

Las acciones que puedo realizar sobre una lista ligada son:

p = Getnode()	Crea un nodo vacío.
GuardaInfo(x)	Guarda información en el sector de DATOS del NODO
GuardaPuntero(p)	Guarda información en el sector de PUNTEROS del NODO.
x = VerInfo()	Muestra la información de DATOS del NODO
p = VerPuntero()	Muestra la información del PUNTERO del NODO

Inserción en una Lista

- Insertar al principio de la lista.
- Insertar al final de la lista.
- Insertar entre 2 nodos de la lista.
 - Esto permite por ejemplo mantener una lista ordenada.
 - Para esto último se debe considerar la búsqueda de un dato en la lista.
 - Ejercicio, crear la función InsertarNodo()

Listas ligadas: eliminar un nodo

- Consideremos el caso de eliminar un nodo al final de la lista.
- Esta operación ahora es muy simple, ya que el último nodo tiene acceso a su predecesor.
- No olvidemos que intentar borrar un elemento de una lista vacía pudiera ocasionar que nuestro programa fracasara, así que habrá que verificar siempre que la lista no esté vacía antes de borrar.
 - Ejercicio, crear la función `EliminarNodo()`

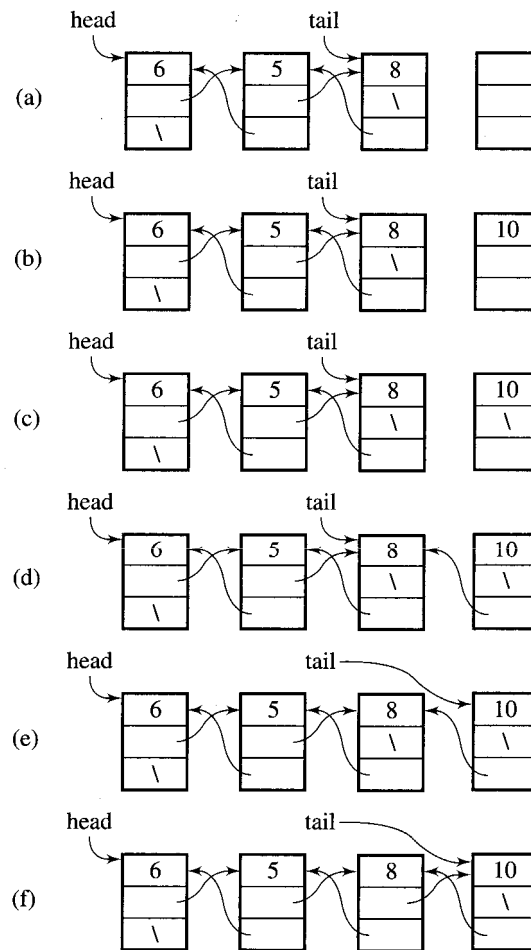
Clasificación de Listas

- Listas simplemente Enlazadas: cada nodo contiene un único enlace que conecta al nodo siguiente o sucesor. (Recorrer hacia adelante)
- Listas doblemente Enlazadas: cada nodo contiene 2 enlaces, uno al nodo siguiente y uno al anterior. (Recorrer hacia delante y hacia atrás)
- Lista circular simplemente enlazada: similar a la primera sólo que el último nodo enlaza con el primero. (Recorrer hacia adelante en forma circular)
- Lista circular doblemente enlazada: similar a la segunda con el último nodo enlazado al primero y el primero al último.

Una lista vacía que no contiene nodos se representa con el nodo “frente” en null.

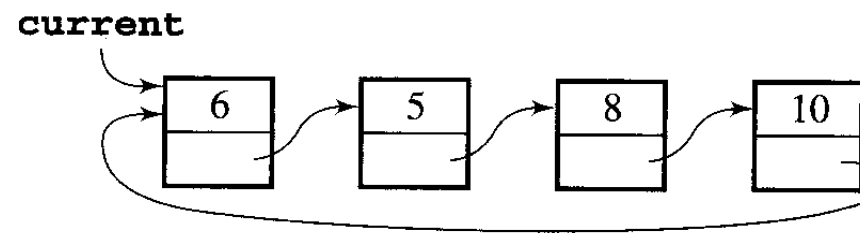
Listas doblemente ligadas: añadir un nodo

- El proceso puede realizarse en los siguientes pasos:
 - Crear un nuevo nodo Fig. (a).
 - Actualizar el value del nuevo nodo Fig (b).
 - Fijar el valor de next en NULL Fig (c).
 - Fijar el valor de prev al valor de tail Fig (d).
 - Tail debe apuntar al nuevo nodo Fig (e).
 - El campo next del predecesor debe apuntar al nuevo nodo Fig (f).
- Note que en el caso de una lista vacia, tanto head como tail apuntan al mismo nodo.



Listas circulares

- En algunas situaciones, una lista circular es necesaria.
- En esta lista finita cada nodo tiene un sucesor.
- Un ejemplo de ello es cuando varios procesos están usando el mismo recurso por la misma cantidad de tiempo y tenemos que asegurarnos de que ningún proceso acceda al recurso antes de que todos los demás lo hayan hecho.



Ejercicios

- Jugar a la rueda (juego de niños).
Objetivo: eliminar competidores, el último que queda gana el juego.