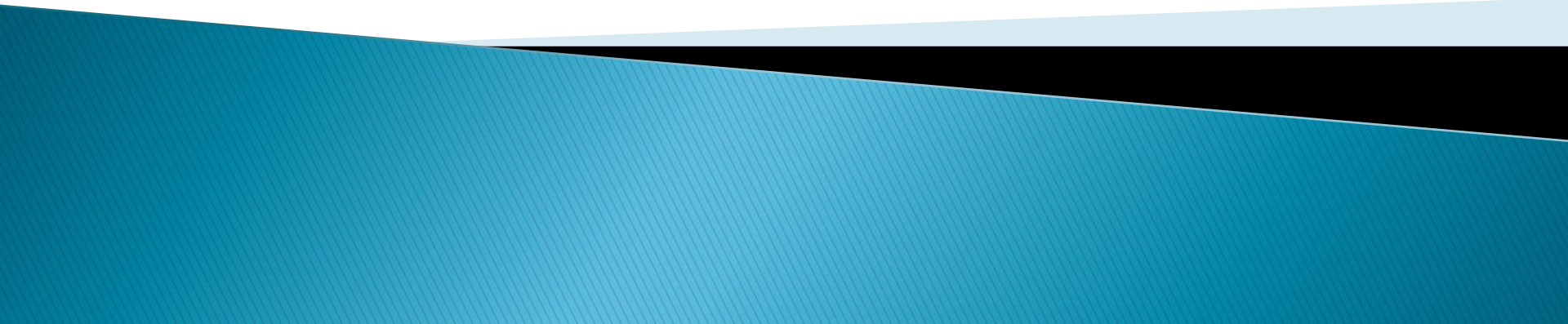


Unidad 3. Estructuras lineales estáticas y dinámicas

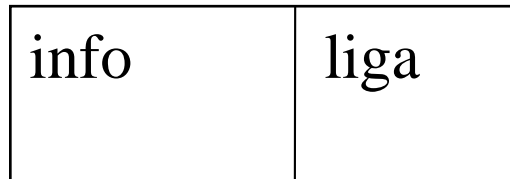


TEMA	SUBTEMA
3 ESTRUCTURAS LINEALES	<ul style="list-style-type: none">3.1 Pilas<ul style="list-style-type: none">3.1.1 Representación en memoria3.1.2 Operaciones básicas3.1.3 Aplicaciones3.2 Colas<ul style="list-style-type: none">3.2.1 Representación en memoria3.2.2 Operaciones básicas3.2.3 Tipos de colas: simples, circulares y bicolos3.2.4 Aplicaciones3.3 Listas<ul style="list-style-type: none">3.3.1 Operaciones básicas3.3.2 Tipos de listas: simplemente enlazadas, doblemente enlazadas y circulares3.3.3 Aplicaciones.

► Listas enlazadas simples

Definición de Lista

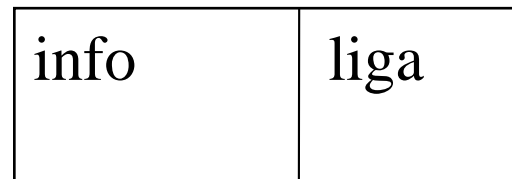
- ▶ Una lista es una colección de elementos llamados nodos. El orden entre los nodos se establece por medio de punteros, es decir, direcciones o referencias a otros nodos.
- ▶ Estructura de un nodo:



- ▶ La lista es un TAD, que consta de una secuencia de elementos llamados nodos.

Nodo

- Info : Datos (Información)
- liga: Enlace o apuntador (Apunta al siguiente nodo)

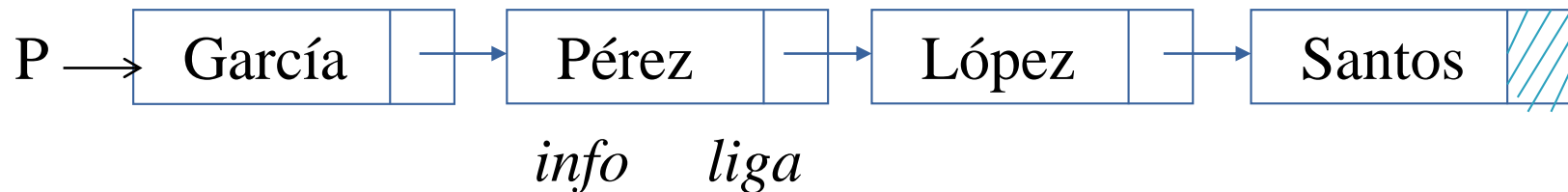


Listas

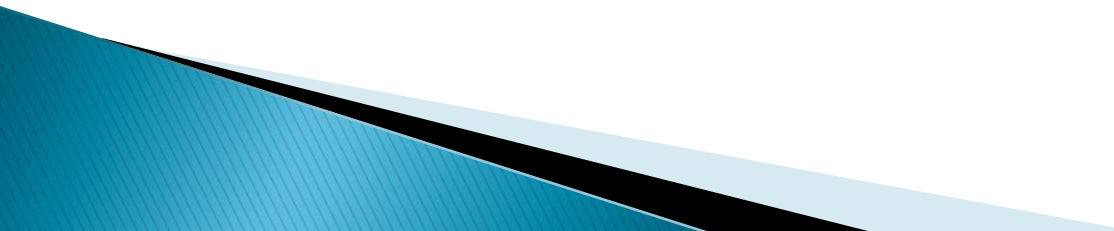
Representación con memoria estática

	<i>info</i>	<i>liga</i>
P → 0	a	1
1	e	2
2	o	-1
3		
4		

Representación con memoria dinámica



Operaciones

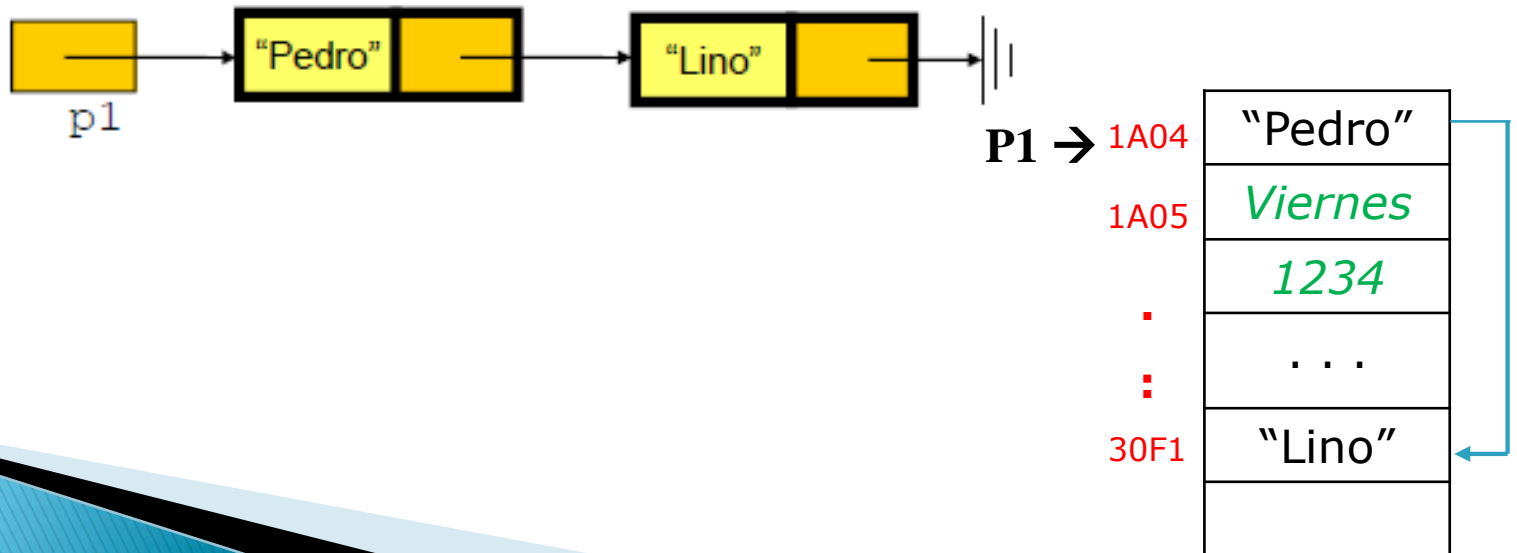
- ▶ Recorrido de la lista
 - ▶ Inserción de un elemento.
 - ▶ Borrado de un elemento.
 - ▶ Búsqueda de un elemento.
- 

- ▶ Listas dinámicas de enlace simple

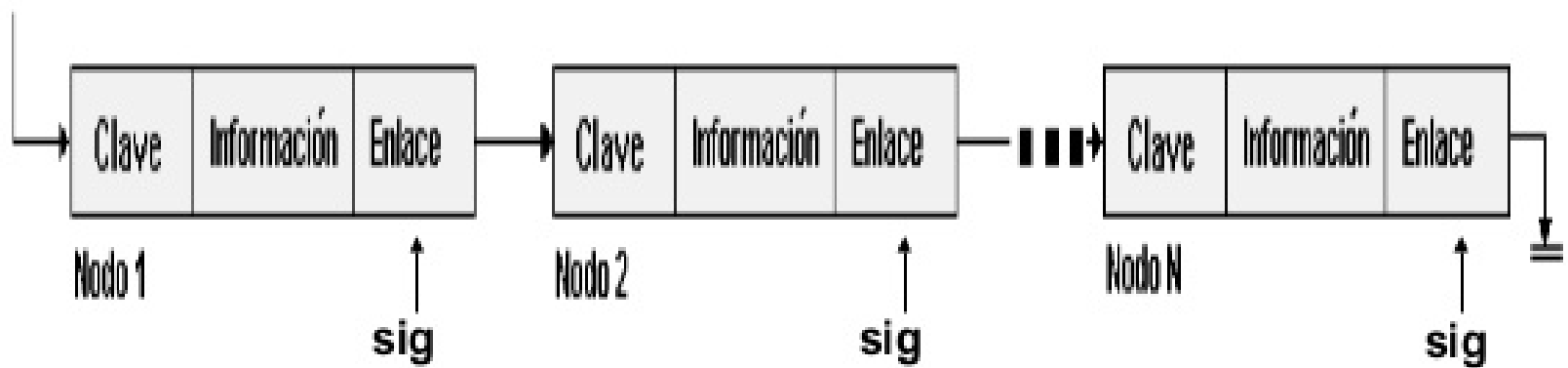
- ❖ Podemos imaginar que en la memoria existirán dos objetos relacionados de la siguiente forma:



- ❖ Representaremos la situación anterior sin hacer explícitas las direcciones de memoria (excepto null):



Comienzo



Operaciones

Recorrido de la lista

crear

insertar

- insertar inicio

- insertar ordenado

- insertar final

eliminar

- elimina inicio

- elimina ultimo

imprimir

modificar

buscar

ordenar

- ordenar ascendente

- ordenar descendente

Inserción de un elemento

- ▶ Consiste en agregar un nuevo nodo a la lista. No se considera el caso de lista vacía, sino que se supondrá que la lista en la cual se inserta el nuevo nodo ya existe.
- ▶ Se pueden presentar tres casos:
 - * insertar un nodo al inicio de la lista.
 - * insertar un nodo al final de la lista.
 - * insertar un nodo antes/después de otro.

InsertalInicio(dato)

{este algoritmo inserta un nodo al inicio de la Lista, P es el apuntador al inicio de la lista, dato es la información que se guarda en el nuevo nodo }

1. Inicio

2. Crea (Q)

3. Hacer $Q.info = dato$, Q

4. $Q.liga = P$,

5. $P = Q$

6. Fin



InsertaFinal(dato)

{Este algoritmo inserta un nodo al final de la lista}

1. Inicio
2. Hacer $\text{aux} = \text{P}$,
3. Repetir mientras $\text{aux.liga} \neq \text{Nulo}$

//recorre la lista hasta llegar al último nodo

hacer $\text{aux} = \text{aux.liga}$

4. Fin del ciclo del paso 3

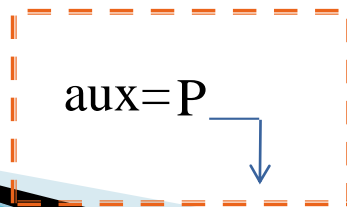
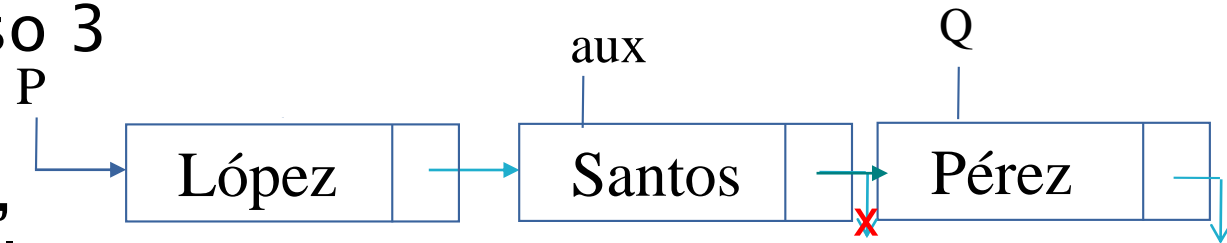
5. Crea (Q)

6. Hacer $\text{Q.info} = \text{dato}$,

7. $\text{Q.liga} = \text{Nulo}$

8. $\text{aux.liga} = \text{Q}$

9. Fin



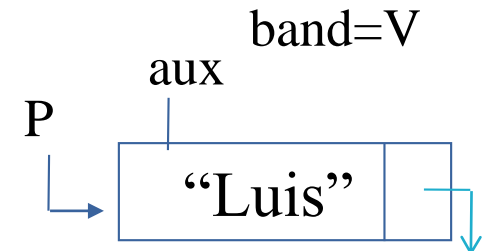
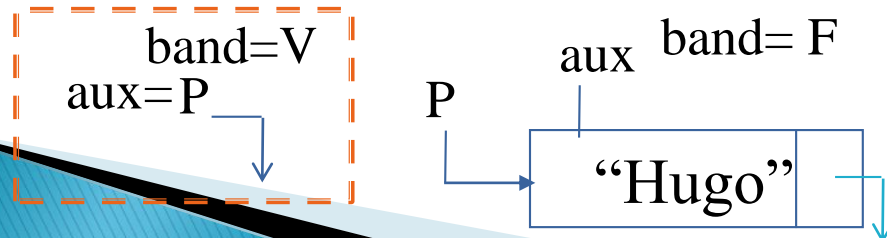
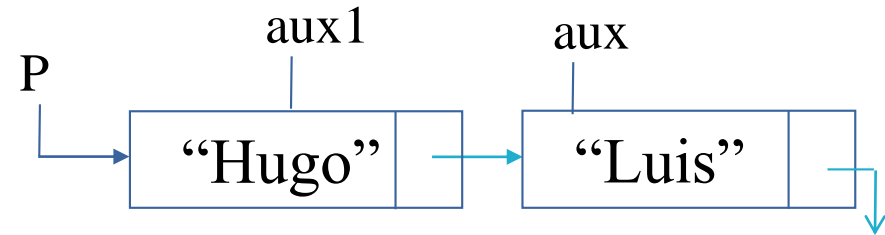
InsertaAntes(dato, Ref)

{ Este algoritmo inserta un nodo antes del nodo Ref }

1. Inicio
2. Hacer $\text{aux} = \text{P}$ y $\text{band} = \text{verdadero}$
3. Repetir mientras $(\text{aux.info} \neq \text{Ref})$ y $(\text{band} = \text{verdadero})$
 - 3.1 Si $\text{aux.liga} \neq \text{Nulo}$
entonces $\text{aux1} = \text{aux}$
 $\text{aux} = \text{aux.liga}$
sino $\text{band} = \text{falso}$
 - 3.2 fin_condicional 3.1
4. Fin del ciclo del paso 3

$\text{dato} = \text{"Paco"}$ $\text{Ref} = \text{"Luis"}$

$\text{band} = \text{V}$



5. Si $band == verdadero$ entonces

Crea(Q)

hacer $Q.info = dato$

5.1 Si $P == aux$ *//es el primer nodo*

entonces $Q.liga = P$

$P = Q$

sino $aux1.liga = Q$

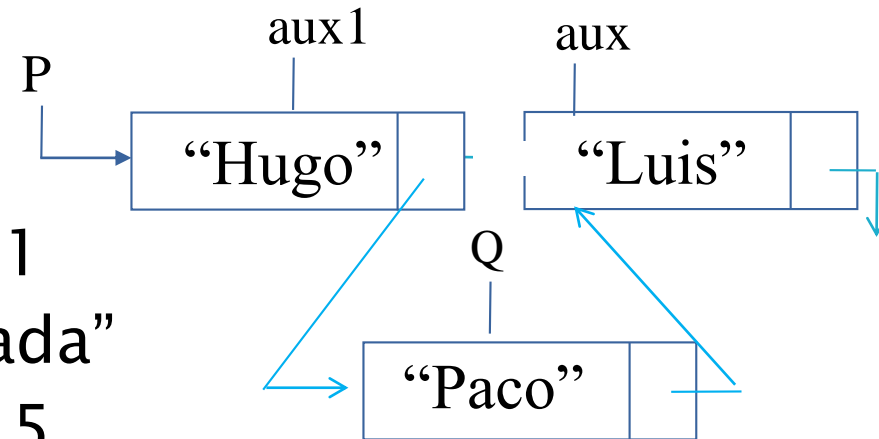
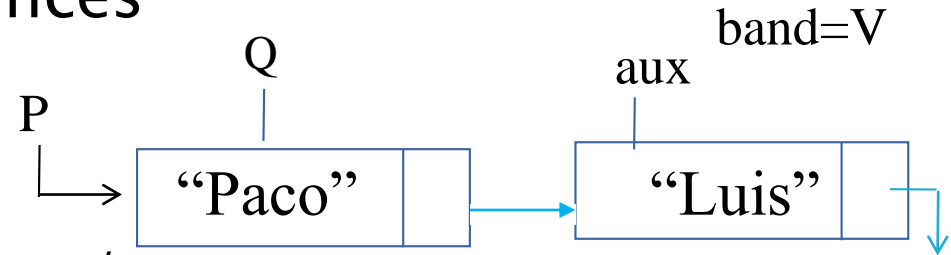
$Q.liga = aux$

5.2 fin del condicional 5.1

sino "Referencia no encontrada"

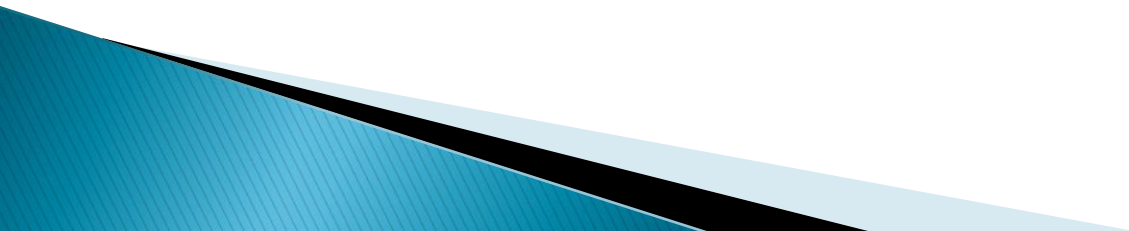
6. Fin del condicional del paso 5.

7. Fin



Ejercicio:

Escribe el algoritmo **InsertaDespues(dato, Ref)**



Borrado de un elemento

- ▶ Consiste en quitar un nodo de la lista, redefiniendo las ligas que correspondan. Se pueden presentar cuatro casos en esta operación:
 - * Eliminar el primer nodo.
 - * Eliminar el último nodo.
 - * Eliminar un nodo con información X.
 - * Eliminar el nodo anterior/posterior al nodo con información X.

EliminaPrimero()

{Este algoritmo borra el primer elemento de una lista. P es una referencia al primer nodo.

Q es una variable de tipo Nodo}

1. Inicio

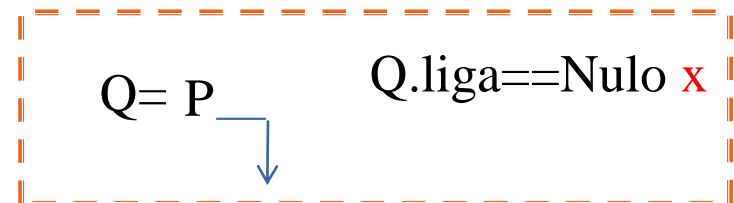
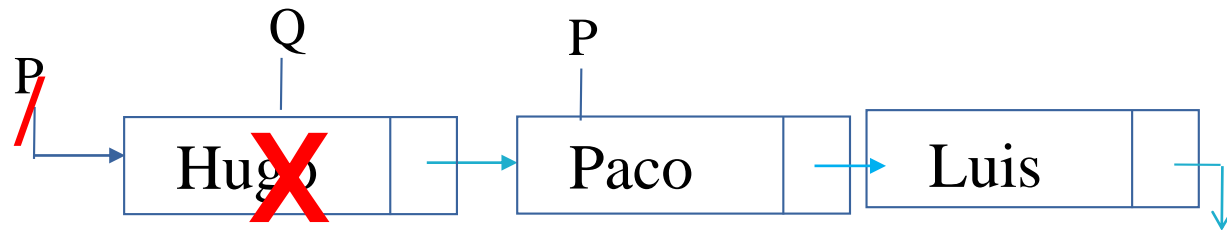
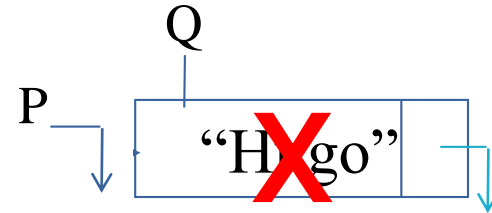
2. Hacer $Q = P$

3. Si $Q.liga == \text{Nulo}$ //verifica si hay un solo nodo
entonces $P = \text{Nulo}$
si no $P = Q.liga$

4 Fin del condicional del paso 3

5. Quita(Q) // libera o destruye nodo Q

6. Fin



EliminaUltimo()

{Este algoritmo borra el ultimo nodo de la Lista P}

1.Inicio

2. Hacer $Q=P$

3.Si $P.liga == Nulo$ //si solo tiene un nodo

$P=Nulo$

sino

3.1 Repetir mientras $(Q.liga \neq Nulo)$

$T=Q$ y $Q=Q.liga$

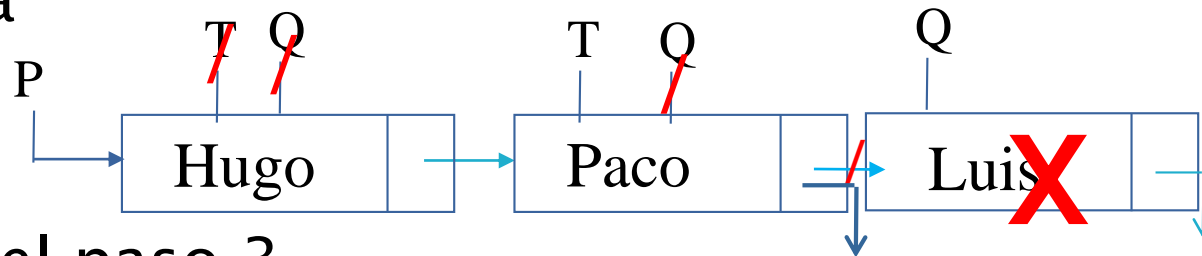
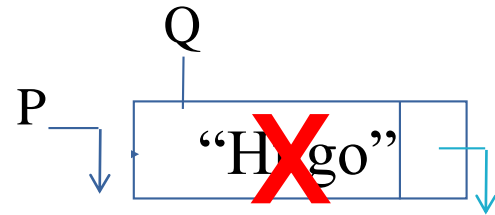
3.2 fin del ciclo 3.1

hacer $T.liga=Nulo$

4.Fin del condicional del paso 3.

5. Quita(Q)

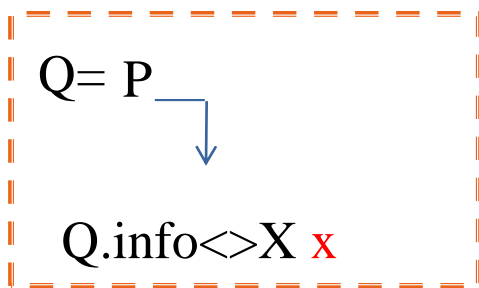
6. Fin



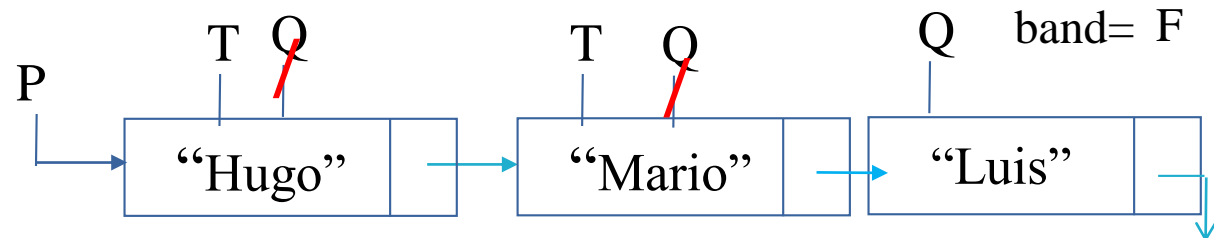
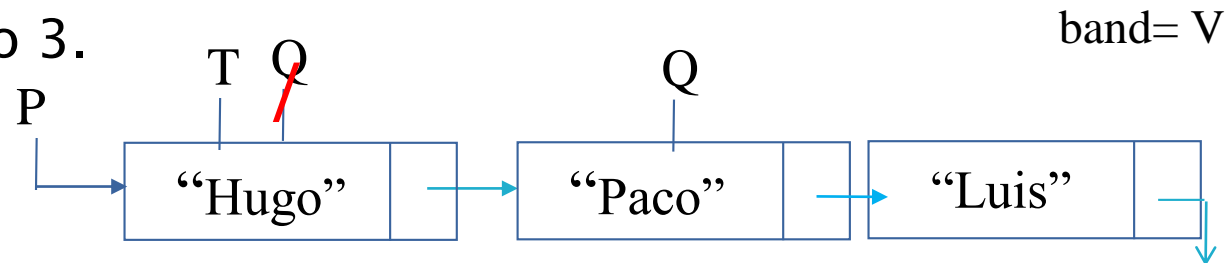
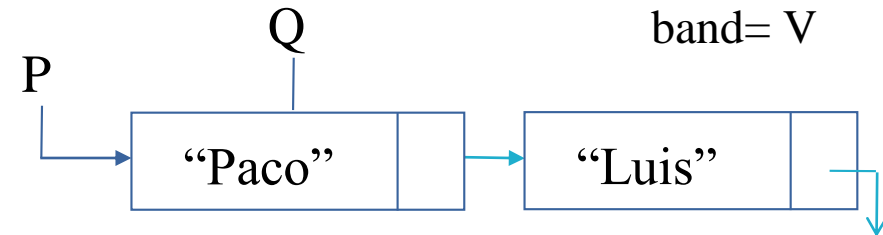
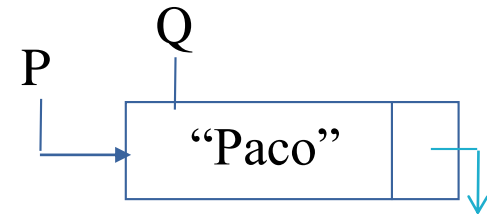
Eliminanodo (X)

{Elimina el nodo con información X.}

1. Inicio
2. Hacer $Q=P$ y $\text{band}=\text{verdadero}$
3. Repetir mientras $(Q.\text{info} \neq X)$ y $(\text{band}=\text{verdadero})$
 - 3.1 Si $Q.\text{liga} \neq \text{Nulo}$
entonces $T=Q$, $Q=Q.\text{liga}$
sino $\text{band}=\text{falso}$
 - 3.2 fin del condicional 3.1
4. Fin del ciclo del paso 3.



$X = \text{"Paco"}$ $\text{band} = V$



5. Si (band=falso)

Escribe(“No existe el valor en la lista”)

sino

5.1 Si (P=Q) //verifica si es el primer nodo

P=Q.liga

sino T.liga=Q.liga

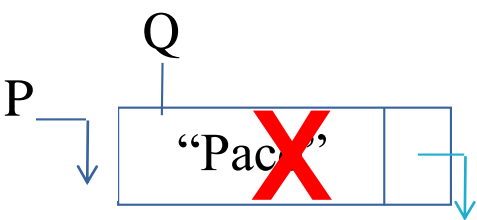
5.2 fin del condicional 5.1

Quitar(Q)

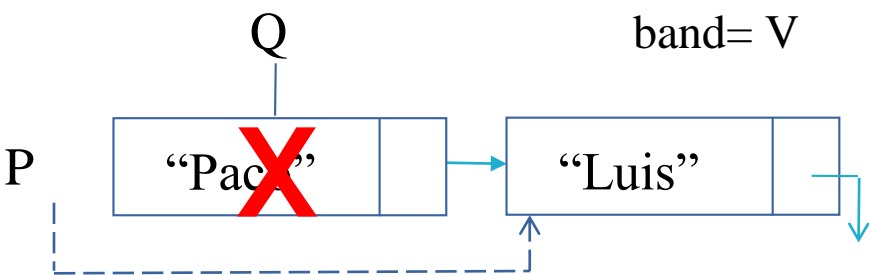
6. Fin del condicional del paso 5.

7. Fin

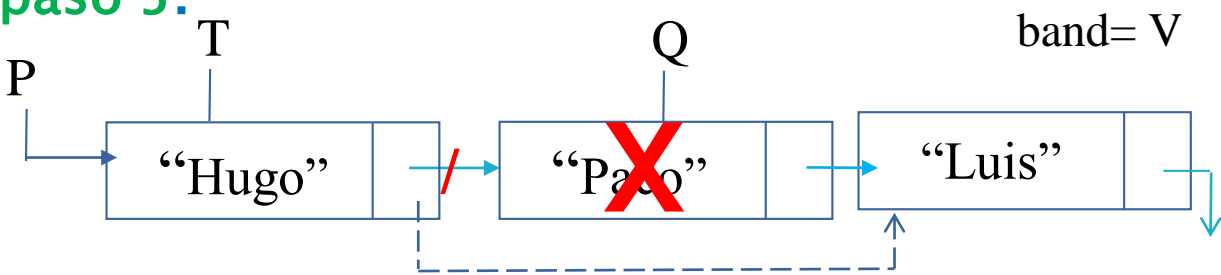
X=“Paco” band= V



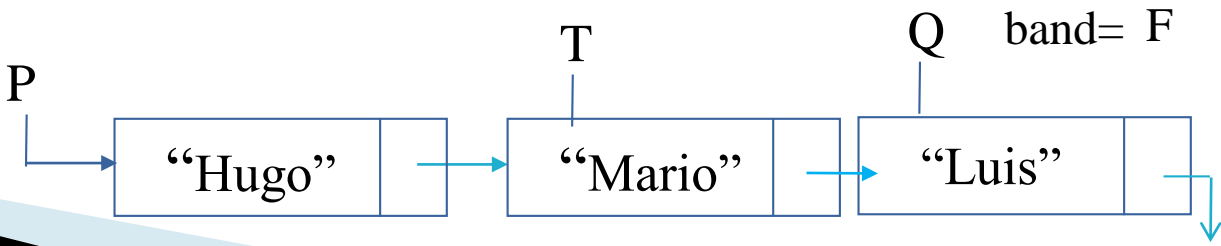
band= V



band= V



band= F



Fuentes de información

- ▶ Estructuras de Datos, Tercera Edición, Osvaldo Cairo y Silvia Guardati, Mc Graw Hill.
- ▶ <http://sabadosinformaticos.blogspot.mx/2013/08/estructura-de-datos-cairo-y-guardati.html>
- ▶ <https://kesquivel.files.wordpress.com/2010/08/listas-circularesv2010.pdf>