Clase #24 de 27
Punteros, Reserva del Heap & Nodos

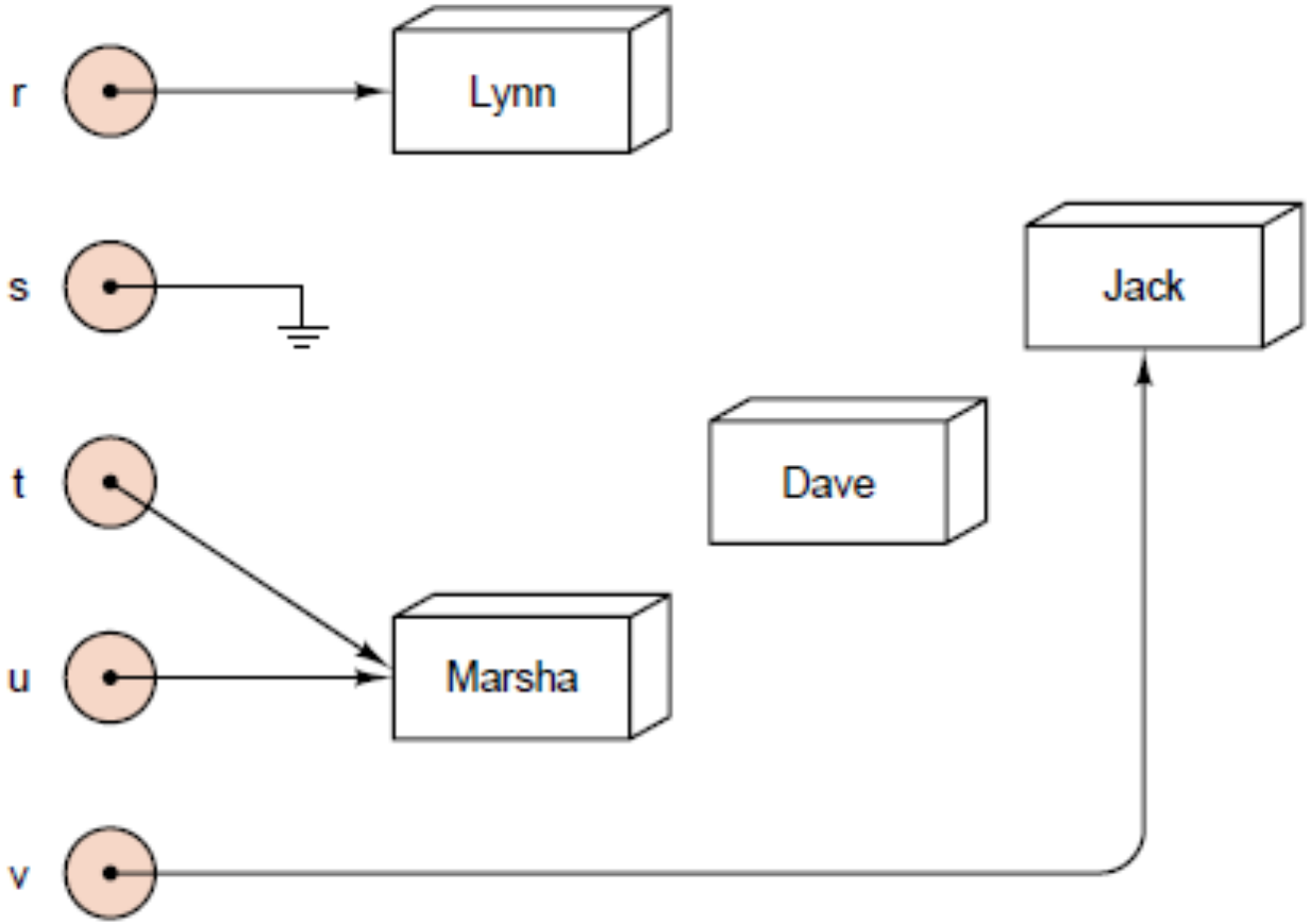
Oct 25, Jueves

Agenda para esta clase

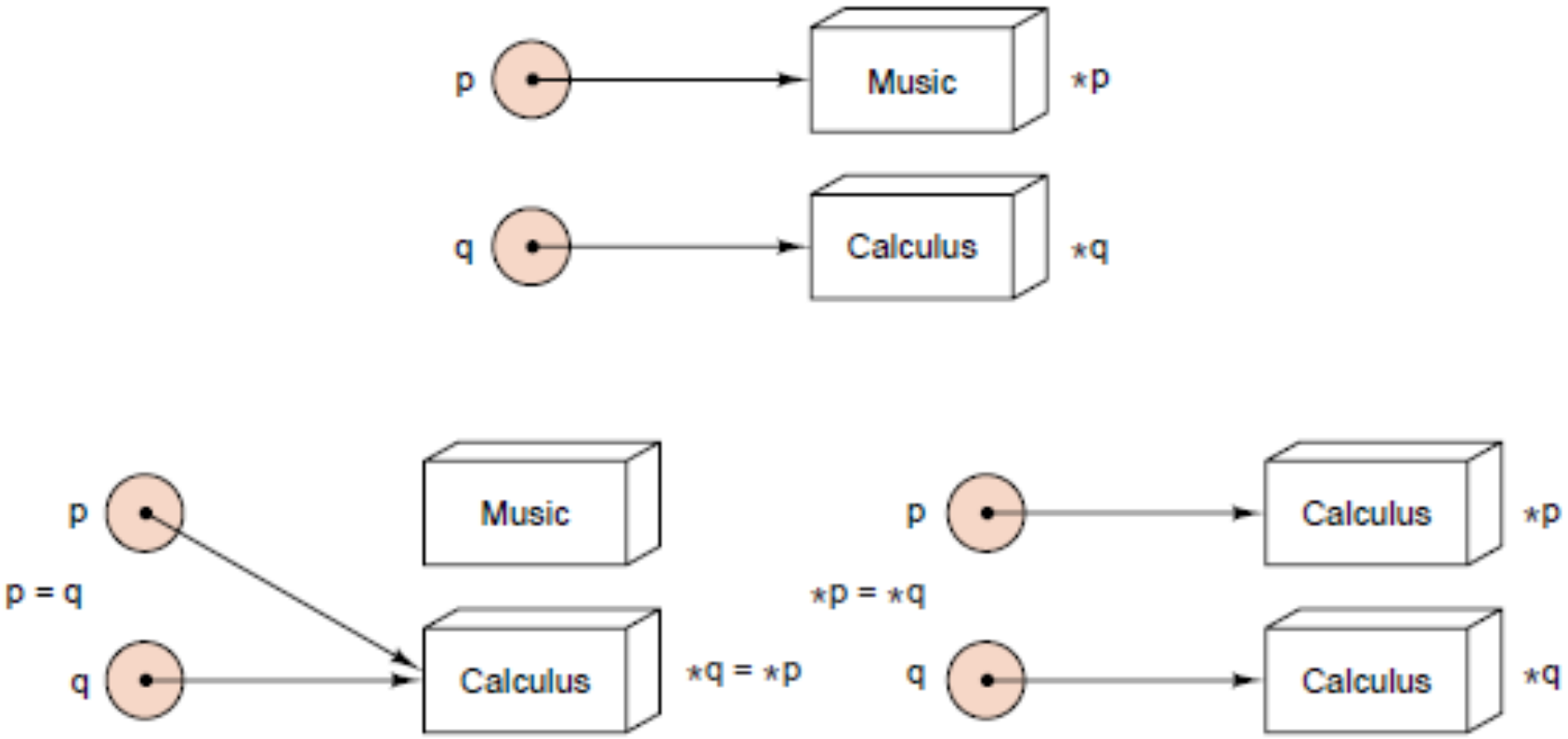
- Punteros
- Reserva Dinámica Manual de Memoria & Heap
- Estructuras Enlazadas & Nodos

Punteros

Punteros



Asignación de Punteros

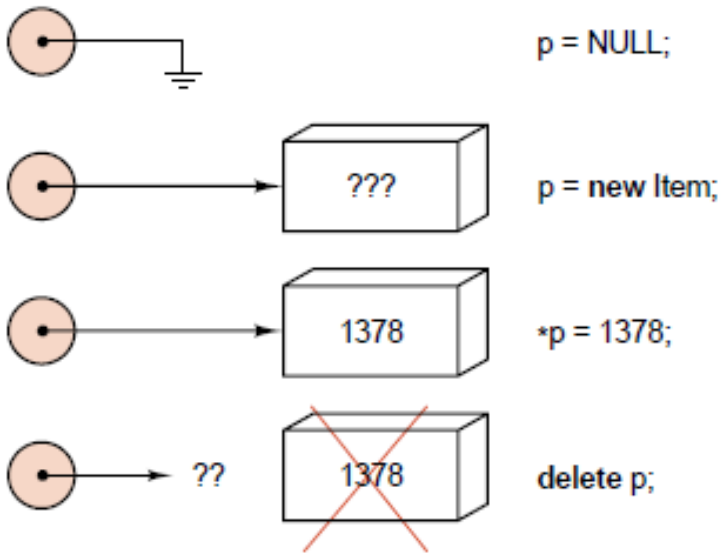


Reserva Dinámica Manual de Memoria & Heap

Reserva Dinámica Manual de Memoria – Heap

```
Type* p;

p = nullptr;

p = new Type;

*p // ??

cin >> *p;

cout << *p;

delete p;
```

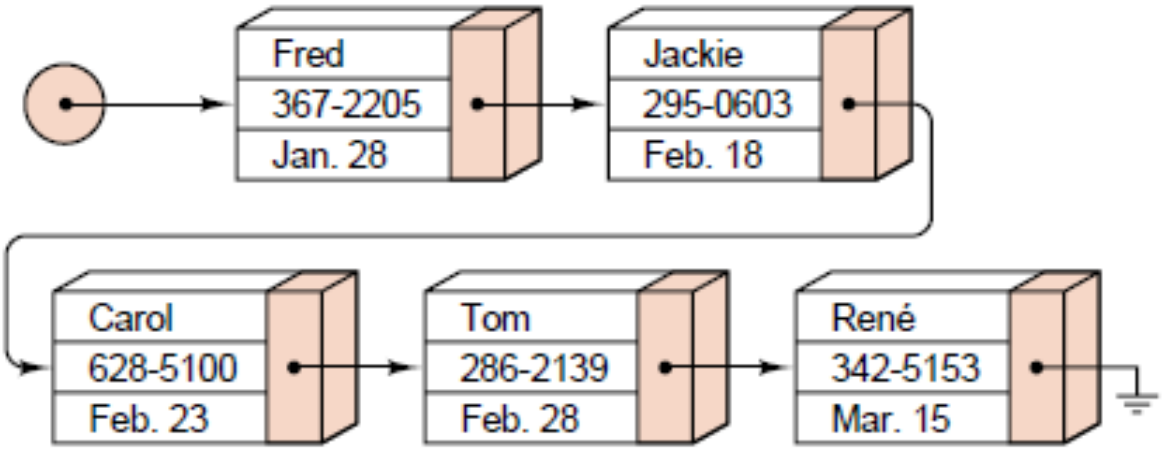
```
int* p; // ??

p // ??

p = nullptr;

p = new Type;

p //

*p // ??

*p = 1378

delete p;
```



- ¿Crea algo nuevo new? ¿Qué?
- ¿Borra algo delete? ¿Qué?

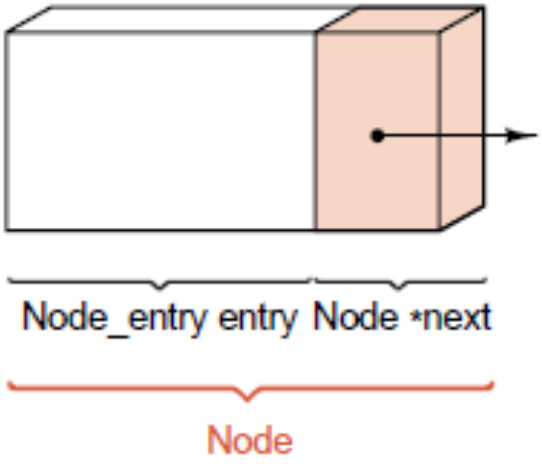
Estructuras Enlazadas & Nodos

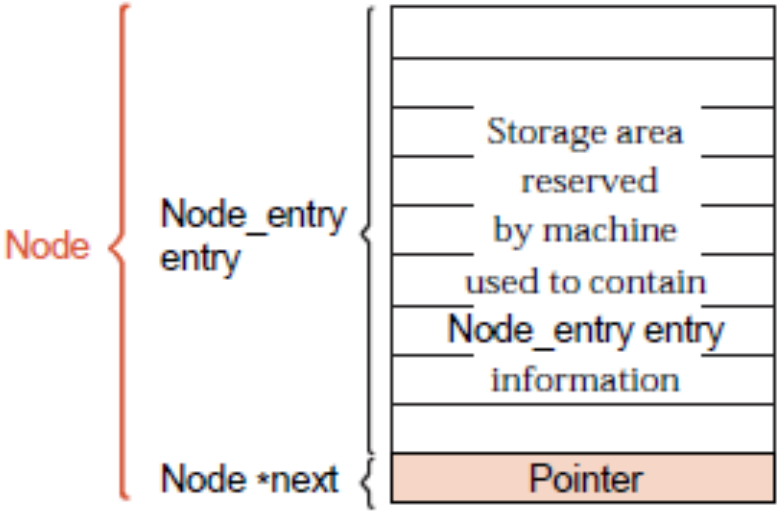
Introducción a Estructura Enlazada



- Repaso: Estructura Estática versus Dináminca
  - Stack, Queue, Array
- Estructura Enlazada
  - Stack, Queue, List
- Problemas que resuelven
- Ventajas y Desventajas

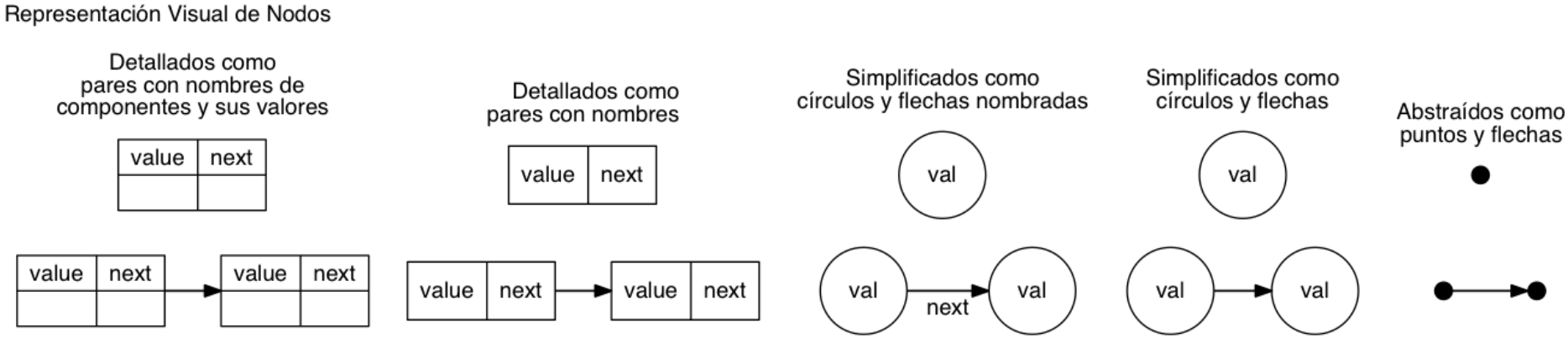
¿Qué es un Nodo? *node*=(*data*,*link*)





- Otros nombres para *data*
  - car
  - datos
  - entry, entrada
  - payload, carga
  - info, información
  - val, **value**, valor
- Otros nombres para *link*
  - cdr
  - enlace
  - **next**, siguiente, próximo
  - nxt, sgte

Representaciones de Nodos

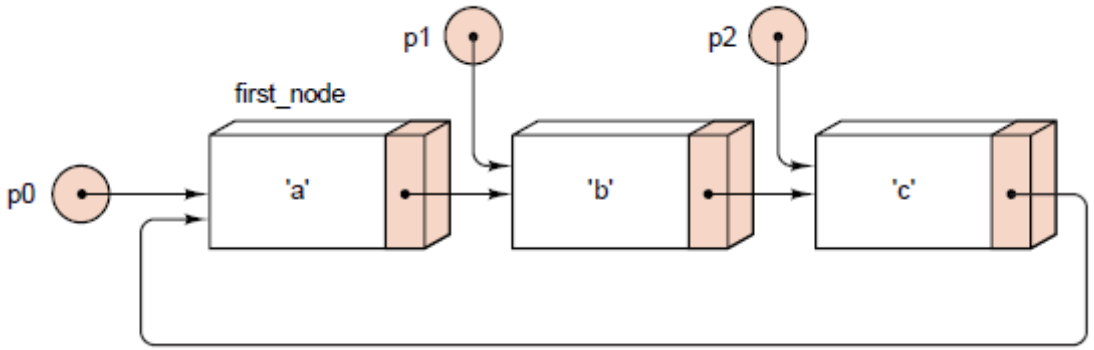



```
// Node of int
struct Node{
   int   value;
   Node* next;
};
```

```
// Node of Type
struct Node{
   Type  value;
   Node* next;
};
```

```
// Template Node
struct Node<T>{
   T         value;
   Node<T>* next;
};
```

Ejemplo: Tres Nodos Enlazados



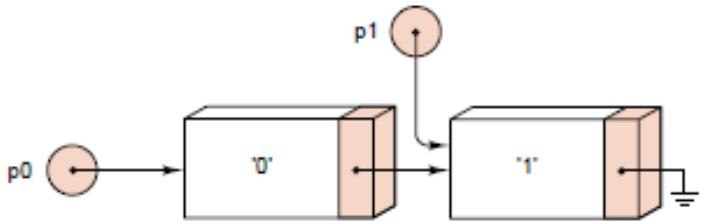

```
Node first_node;
first_node.value = 'a';

Node* p0 = &first_node;
Node* p1 = new Node;

(*p1).value = 'b';
p1->value = 'b';

p0->next = p1;

Node *p2 = new Node;
p2->value = 'c';
p2->next = p0;

p1->next = p2;

// Recorrido
Node *p=p0; // actual
do{
  cout << p->value;
  p=p->next;
}while( p != p0);
```

Ejercicio – Dibuje el diagrama para el siguiente código C++

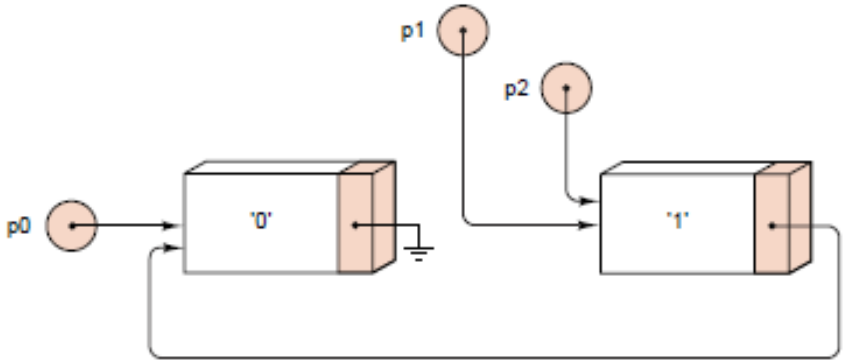
```
Node* p0 = new Node;
po->value ='0';

Node* p1 = p0->next = new Node;
p1->value ='1';

Node* p2 = p1->next = new Node;
p2->value ='2';
p2->next = p1;
```

Ejercicios – Escriba las sentencias

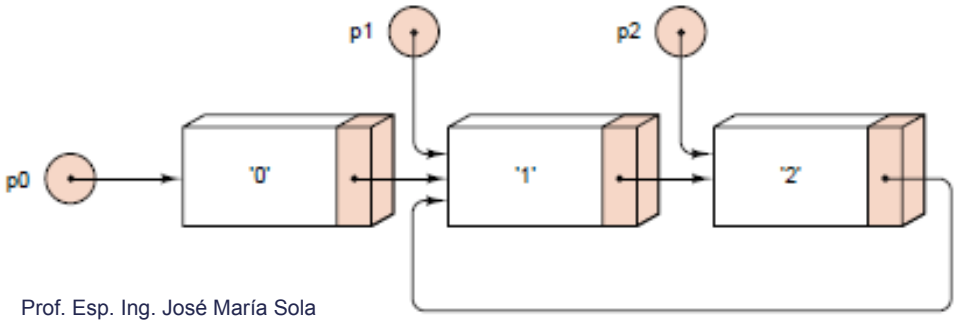
(a)



(b)



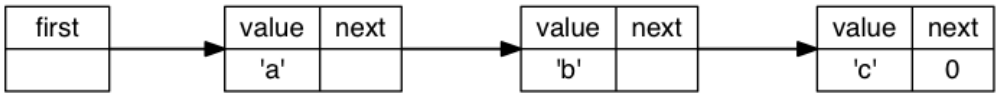
(c)



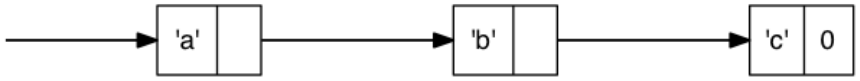
Recorrer la lista enlazada

Representación Visual de secuencia de caracteres (a,b,c) como lista enlazada

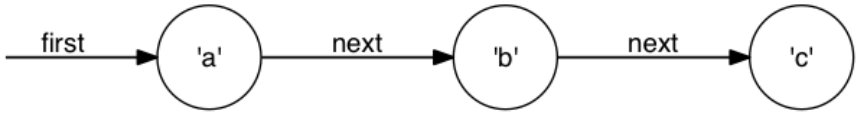
Detallados como pares



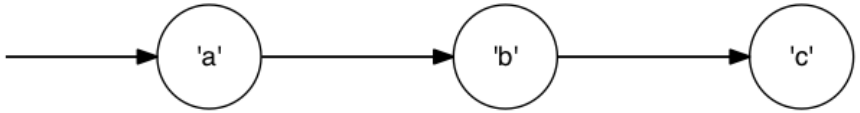
Detallados como pares sin nombres



Simplificados como círculos y flechas nombradas



Simplificados como círculos y flechas



Abstraídos como puntos y flechas

```
Node* first = …;

for(Node* p=first; p; p=p->next)
  cout << p->value;
```

¿Consultas?

Fin de la clase