

Estructuras de Datos

Profesor

Sergio Gonzalez

Unidad 6: Listas

Profesor

Sergio Gonzalez

Estructuras estáticas

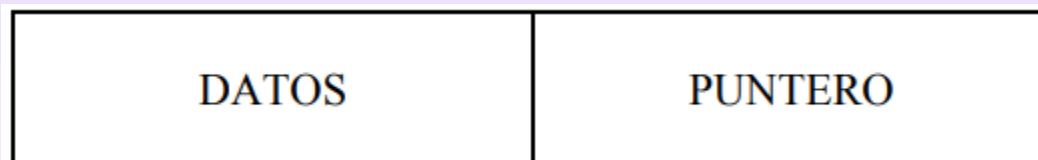
- Hasta ahora trabajamos con estructuras estáticas
- Posiciones de memoria específicas
- Variables perfectamente dimensionadas

Estructuras dinámicas

- Adquirir posiciones de memoria adicionales
- Variables que se dimensionan (crecen y decrecen) en tiempo de ejecución
- Punteros → Posición de memoria

Estructuras dinámicas

- Punteros → Enlaces entre elementos de la estructura
- Así las estructuras pueden variar su tamaño
- Elementos de estructura → Nodos



Estructuras dinámicas

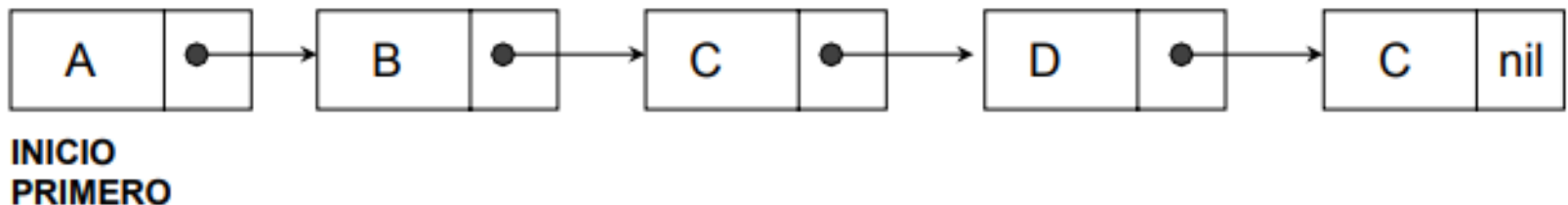
- Dependiendo de las relaciones entre nodos:
 - Estructuras lineales
 - Listas
 - Pilas
 - Colas
 - Estructuras no lineales:
 - Árboles
 - Grafos

Listas

- Conjunto de datos de un mismo tipo (??)
- Cantidad variable
- Cada elemento tiene un único predecesor y un único sucesor
 - Excepto el primero y el último

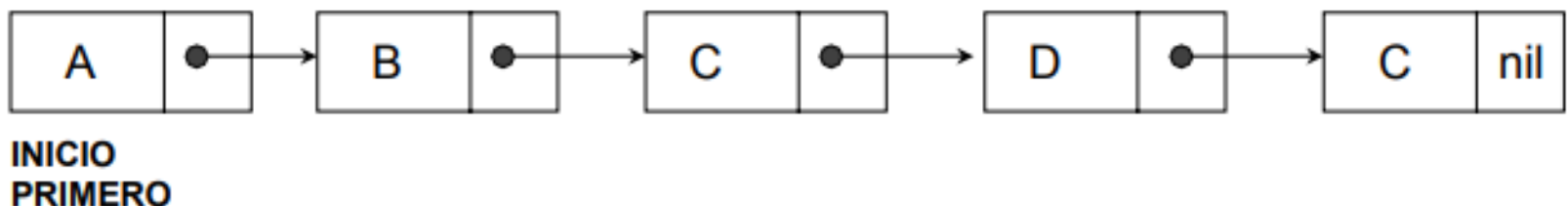
Listas simples enlazadas

- Cada nodo contiene:
 - Dato (información)
 - Enlace al siguiente nodo
- Nodo Inicial → Nombre de lista
- Nodo final → Enlace vacio



Listas simples enlazadas

- Creación de una lista:
 - Crear lista vacía
 - Enlace vacío
- A partir del nodo inicial:
 - Modificación, crecimiento, inserción, decrecimiento, borrado



Listas simples enlazadas

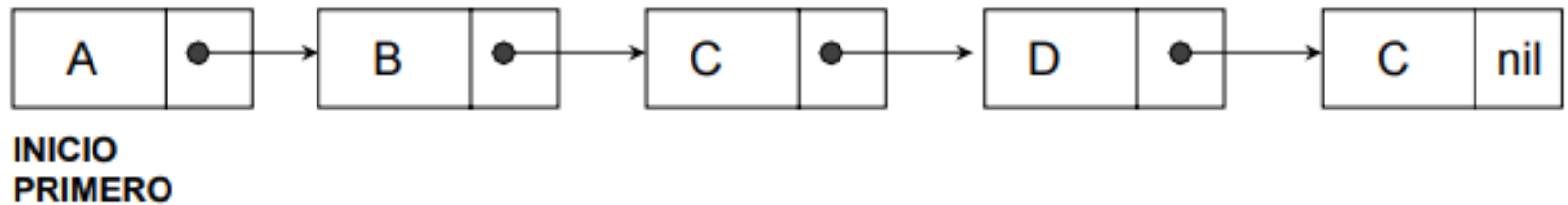
- Operaciones
 - Crear
 - Recorrido
 - ObtenerElemento
 - Tamano
 - EstaVacia
 - EstaLLena
 - EliminarLista
 - Insertar elemento
 - Agregar
 - Insertar
 - Eliminar elemento

Listas simples enlazadas

- Crear
 - Generar lista vacia
 - Sin ningún nodo

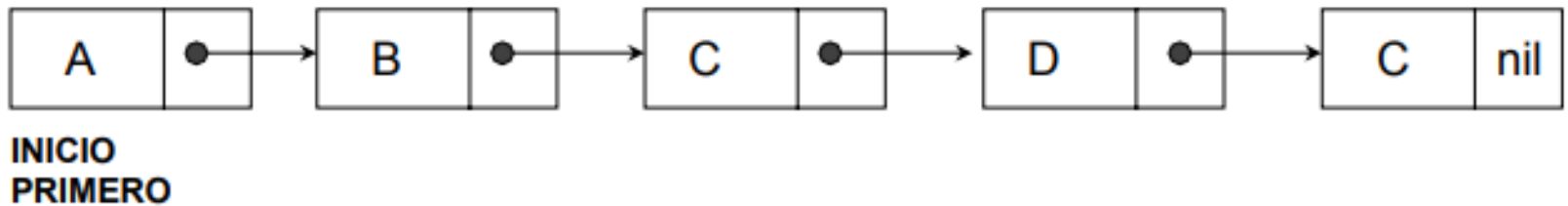
Listas simples enlazadas

- Recorrer
 - Pasar por los nodos de la lista hasta el final



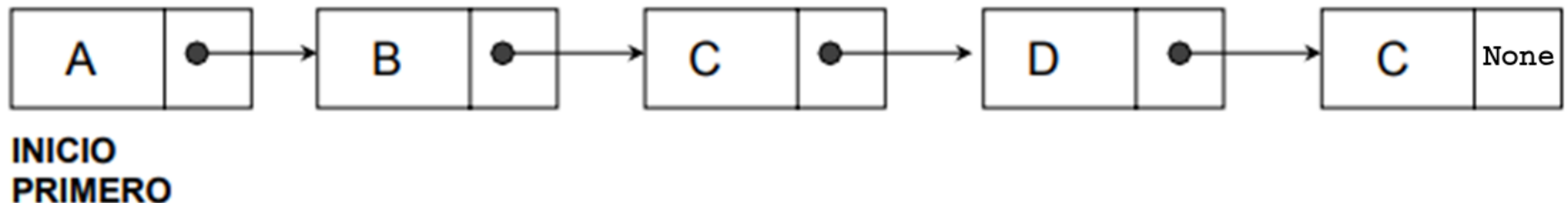
Listas simples enlazadas

- ObtenerElemento
 - Obtener el dato de una posicion determinada de la lista (nodo en una posicion)



Listas simples enlazadas

- Tamaño
 - Recorrer lista sumando en un contador para obtener la cantidad de nodos.



Listas simples enlazadas

- Insertar al final (Agregar – Append)
 - Se crea nuevo nodo
 - Si la lista esta vacia:
 - Se agrega nuevo nodo
 - Si no esta vacia:
 - Se recorre la lista hasta el final
 - Se agrega nuevo nodo

Listas simples enlazadas

- Insertar en una posición específica (incluye inicio)
 - Se crea nuevo nodo
 - Se recorre lista hasta la posición deseada:
 - Se agrega nuevo nodo
 - Si la posición deseada es mayor que la cantidad de elementos:
 - Se agrega nuevo nodo al final

Listas simples enlazadas

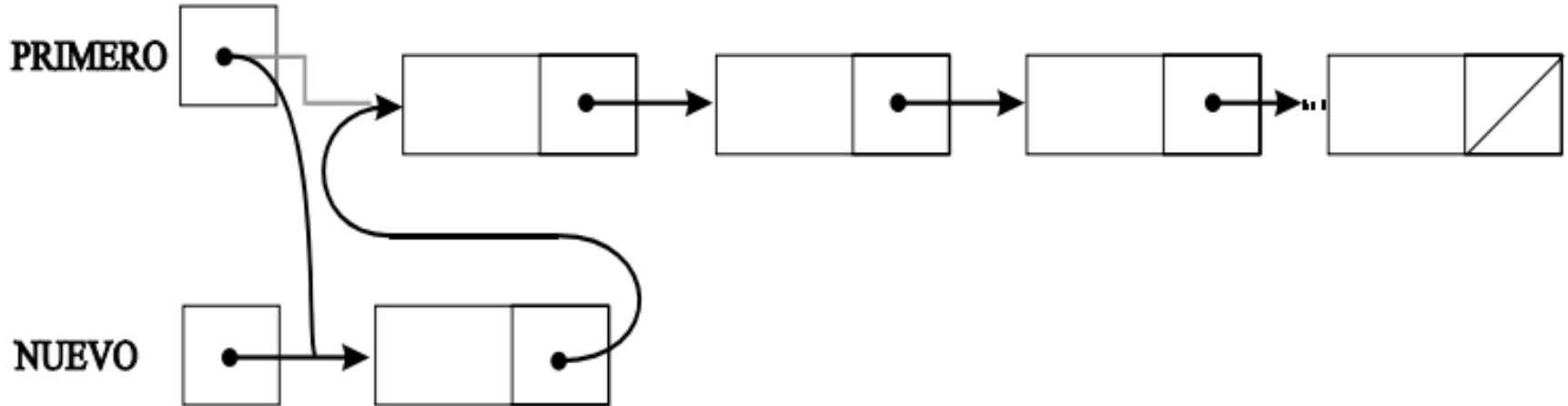
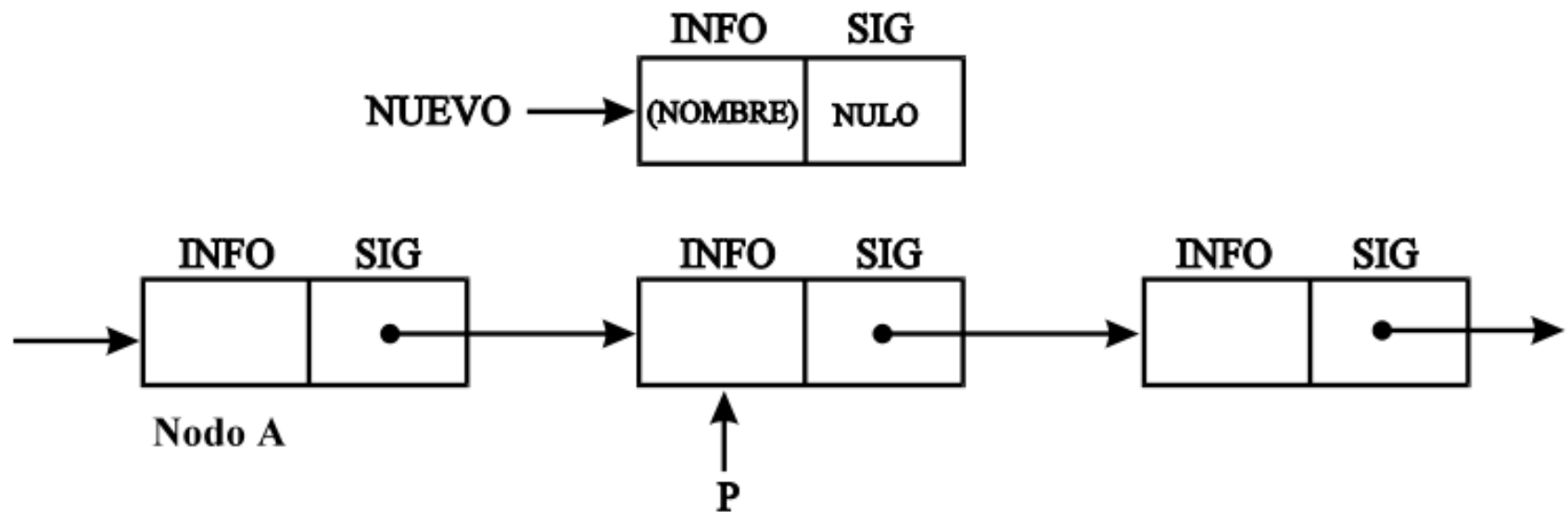


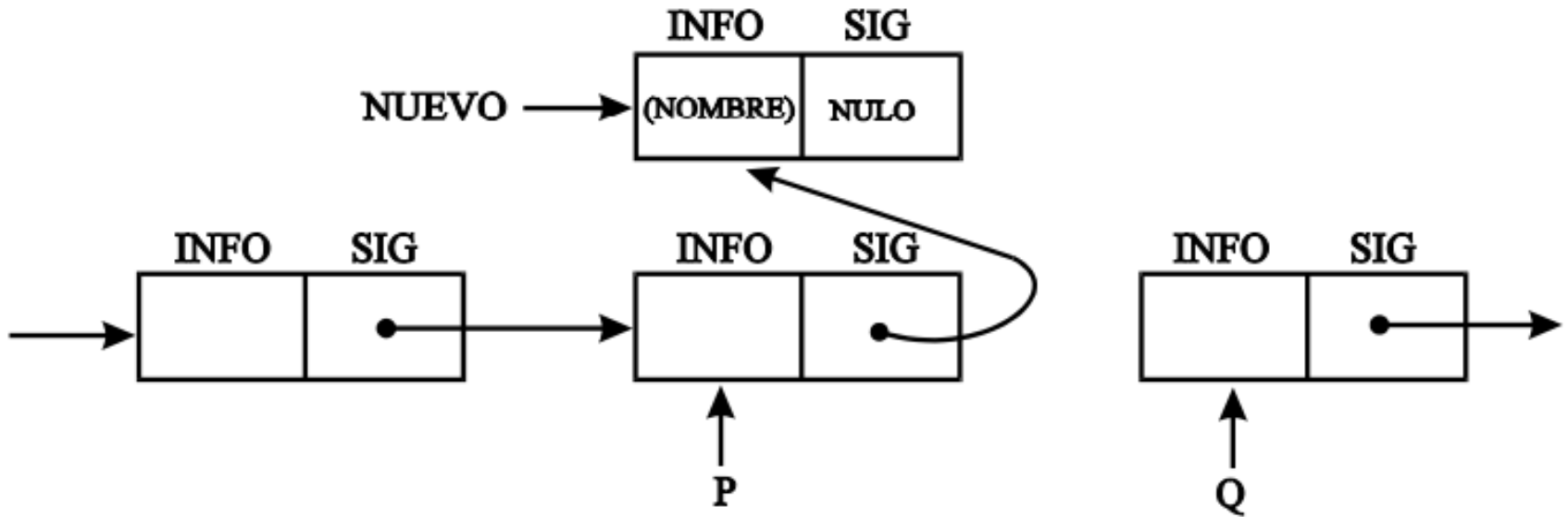
Fig. 5.7. *Insertión en el primer nodo*



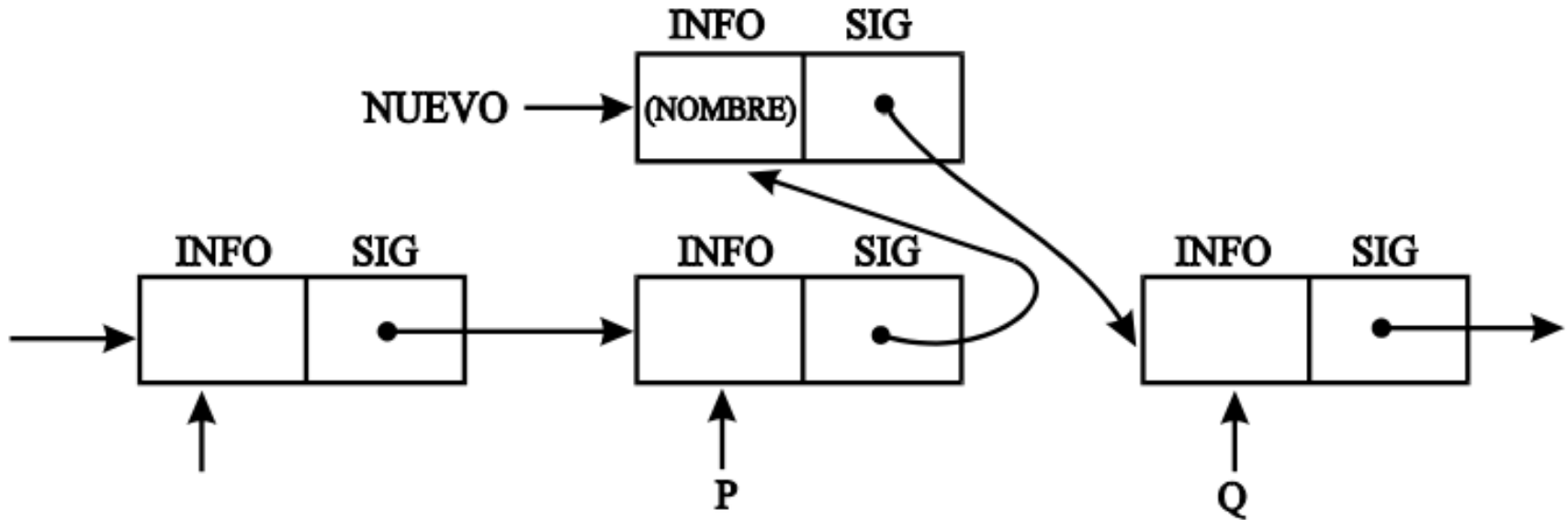
Listas simples enlazadas



Listas simples enlazadas



Listas simples enlazadas



Listas simples enlazadas

- Eliminar de una posición específica (incluye inicio y final)
 - Un nodo quedará fuera de la lista
 - Se recorre lista hasta la posición deseada:
 - Se enlaza nodo anterior con nodo siguiente y el actual queda fuera de la lista

Listas simples enlazadas

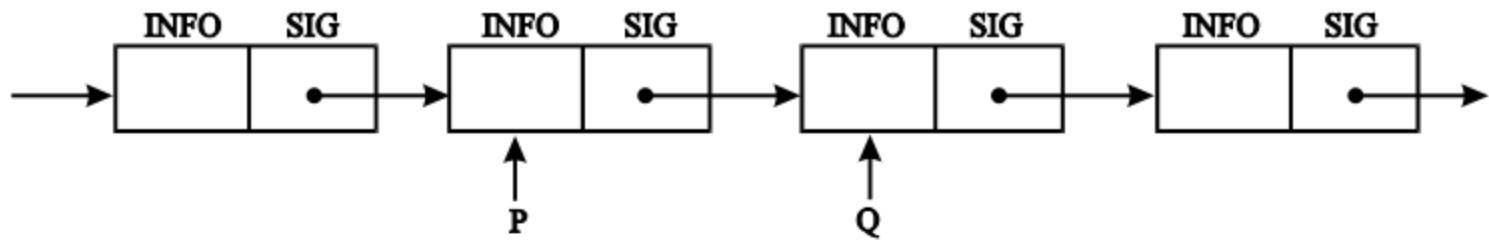


Fig. 5.9 (b) *Situación tras el paso 1*

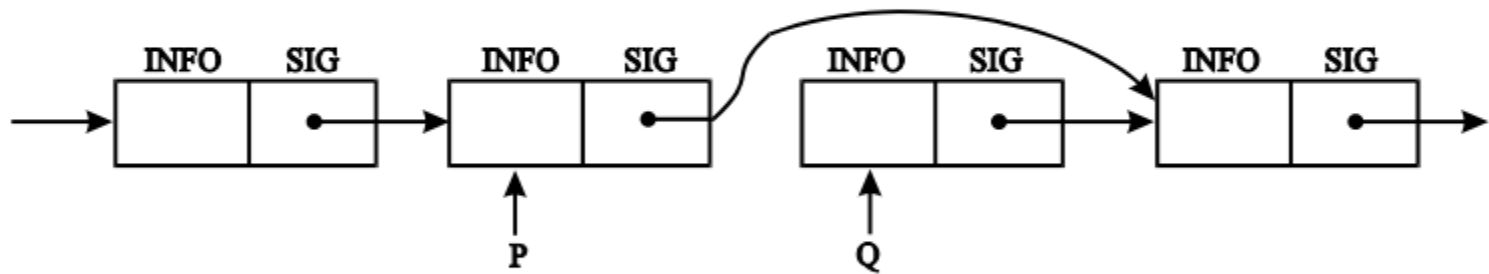
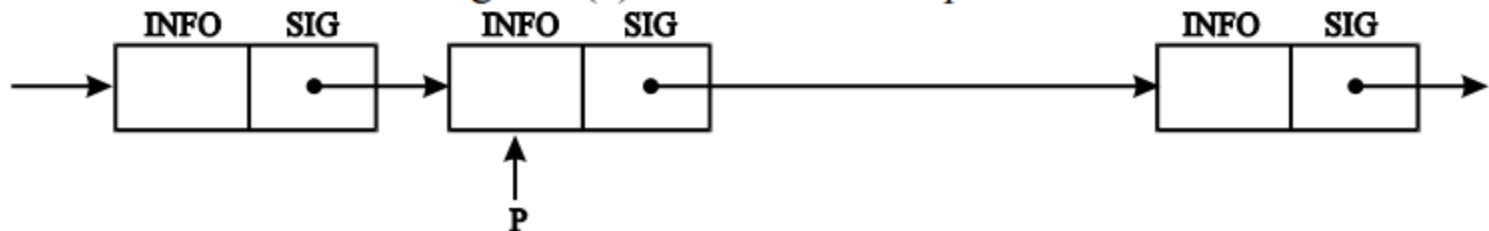


Fig. 5.9 (c) *Situación tras el paso 2*



Listas simples enlazadas

- ¿Como podemos implementarlo en Python?
 - TDA NodoLista
 - TDA Lista

Listas simples enlazadas

- ¿Como podemos implementarlo en Python?
 - Una unica referencia al inicio
 - Dos referencias: Inicio y Final

Listas simples enlazadas

- ¿Como podemos implementarlo en Python?

```
class NodoLista:
```

```
    def __init__(self, dato = None):  
        self.dato = dato  
        self.siguiente = None
```

Listas simples enlazadas

- ¿Como podemos implementarlo en Python?

```
class Lista:
```

```
    def __init__(self):  
        self.primeros = None
```

Listas simples enlazadas

- ¿Como podemos implementarlo en Python?

class Lista:

```
    def recorrido(self):
```

```
        aux = self.primer
```

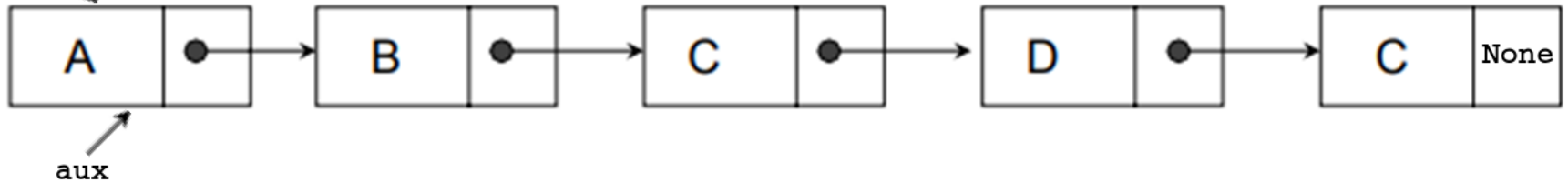
```
        while aux != None:
```

```
            aux = aux.siguiente
```

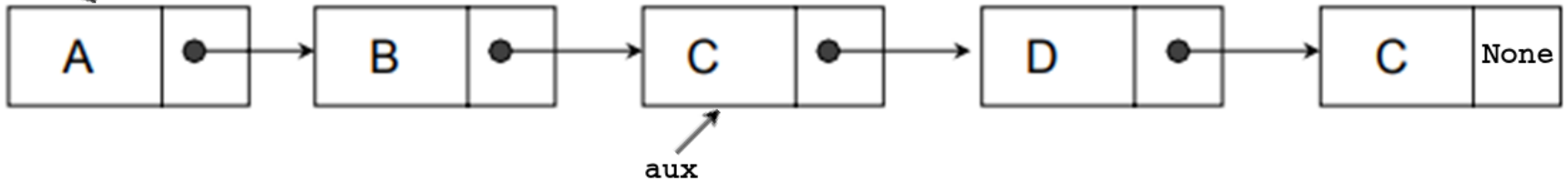


Listas simples enlazadas

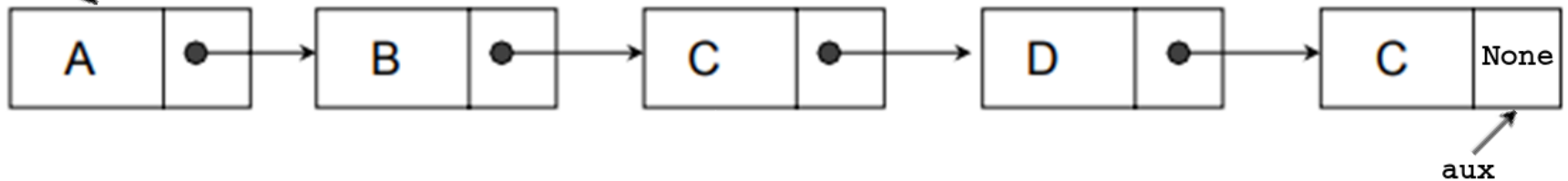
`self.primer`



`self.primer`

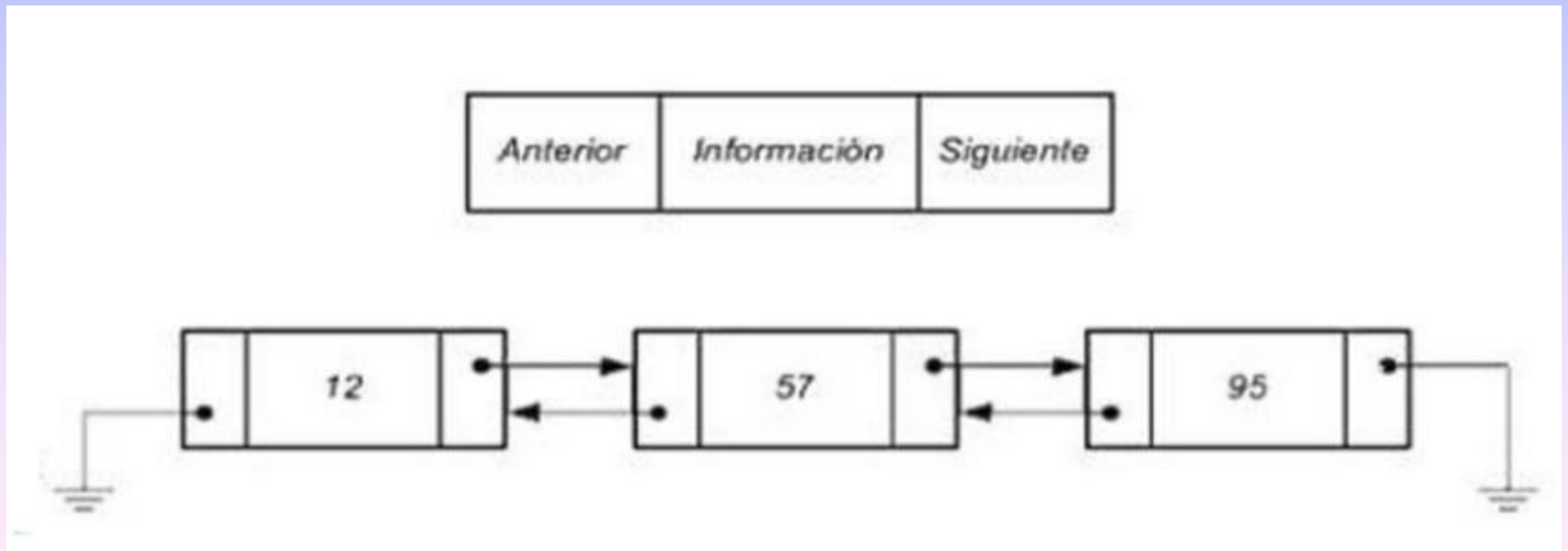


`self.primer`



Listas dobles enlazadas

- Dos enlaces en cada nodo: Anterior y siguiente nodo



Listas dobles enlazadas

- Elemento anterior primero es null
- Elemento siguiente al ultimo es null
- Podemos movernos desde un nodo hacia el anterior o el siguiente
- Se puede recorrer en ambos sentidos

Listas circulares

- Simples
- Dobles
- Ultimo nodo conectado con el primero
- No hay ningun enlace null

