

# **TIPOS DE DATOS ABSTRACTOS EN EL LENGUAJE C++**

Profesor: Mg. Miguel Guanira

# ESTRUCTURAS AUTOREFERENCIADAS EN EL LENGUAJE C++

# Definición:

Una estructura autoreferenciada, son estructuras que pueden enlazarse o conectarse, por medio de punteros, a otras estructuras similares, de modo que podamos formar con ellas conjuntos de elementos que empiecen sin elementos y durante la ejecución del programa este conjunto pueda crecer o reducirse.

Por medio de estas estructuras y de acuerdo a la forma cómo se enlacen, se podrán crear “listas simplemente ligadas”, “listas doblemente ligadas”, “árboles”, “pilas”, “colas”, etc.

# Implementación:

La forma más básica de estructura, se define de la siguiente manera:

```
struct Nodo {  
    Tipo dato;  
    struct Nodo *siguiente;  
}
```

Donde:

**Tipo**: es cualquier tipo de datos sin distinción, dependerá de lo que se quiera guardar en el conjunto.

**siguiente**: es un puntero que puede apuntar a una variable del tipo **struct Nodo**.

# Interpretación:

Una estructura autoreferenciada se maneja necesariamente con memoria dinámica.

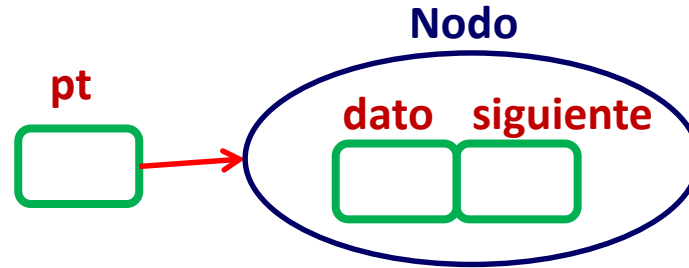
En ese sentido un programa que quiera utilizarla definirá un puntero de la siguiente forma: `struct Nodo *pt = nullptr;`

Como se ve, no existen datos inicialmente.

Luego podemos crear un elemento con la siguiente instrucción:

```
pt = new struct Nodo;
```

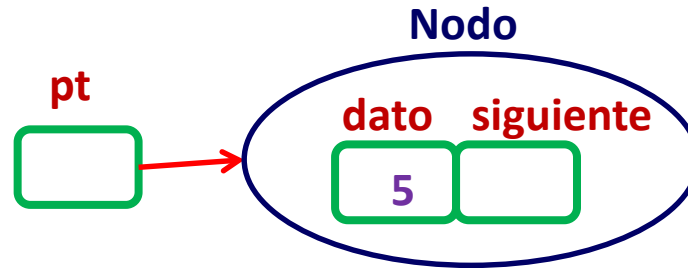
Con esta orden tendríamos el siguiente esquema:



Luego podríamos empezar a llenar los campos:

```
pt->dato = 5; //suponiendo que Tipo sea int
```

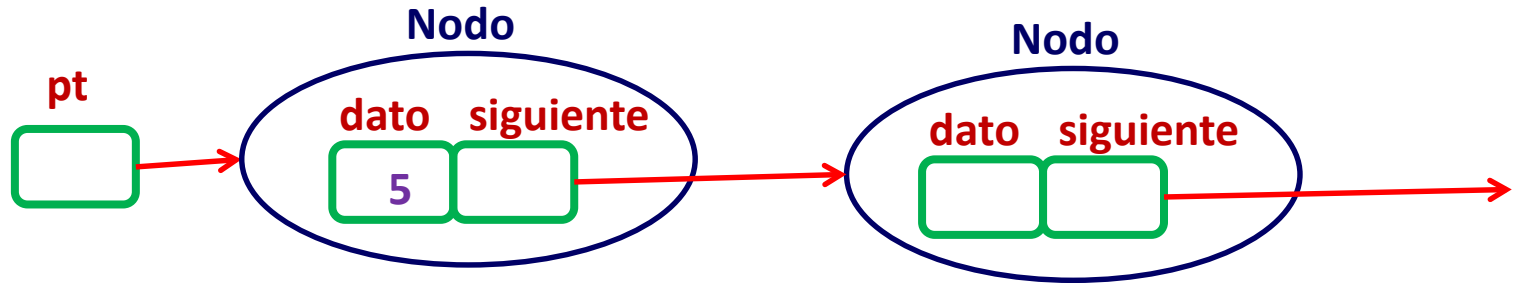
Entonces tendríamos:



Pero también podríamos asignar el otro campo:

```
pt->siguiente = new struct Nodo;
```

Ahora el esquema que tendríamos es el siguiente:

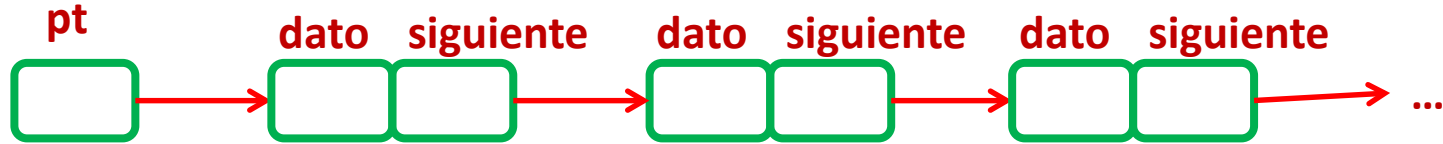


Como se puede ver, la estructura no tenía inicialmente datos, después tuvo un dato y ahora tiene dos.

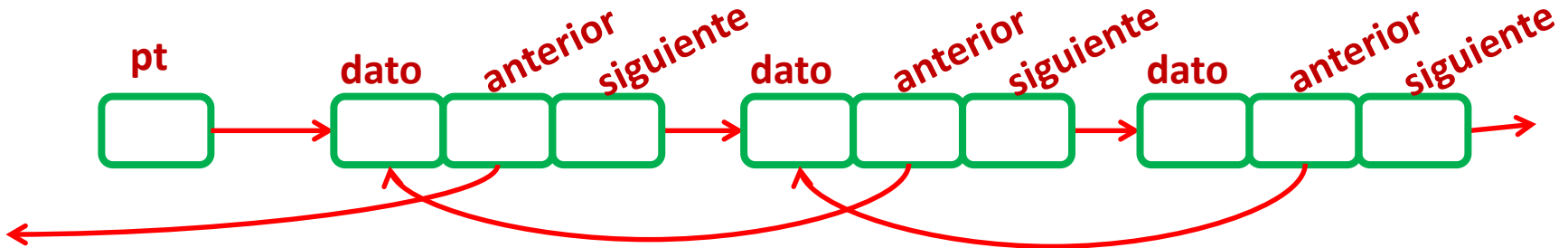
Con algoritmos convenientes podemos hacerla crecer o reducirse según las necesidades..

# Tipos de estructuras:

## 1. – Lista simplemente ligada



## 2. – Lista doblemente ligada





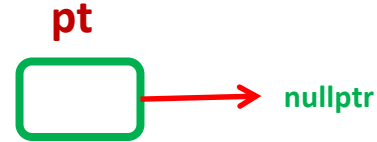


# Tipos de estructuras:

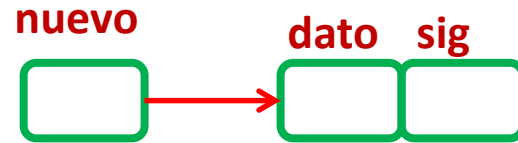
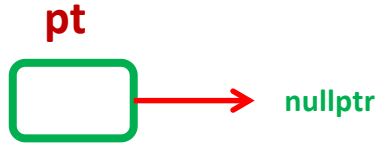
- 4. – Pilas, son listas ligadas en las que los datos se colocan y sacan por el mismo lugar de modo que el ultimo en llegar es el primero en salir. Se denominan estructuras LIFO (Last In, First Out).
  
- 5. – Colas, son listas ligadas en las que los datos se colocan por un extremo y se sacan por el otro. Se denominan estructuras FIFO (First In, First Out).

# Algoritmo para crear una lista ligada simple:

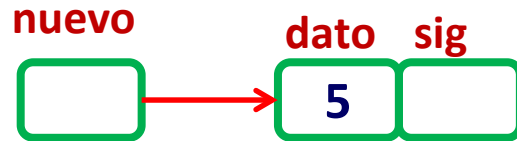
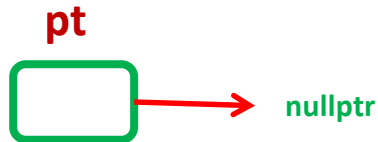
1. – `struct Nodo *pt = nullptr, *nuevo;`



2. – `nuevo = new struct Nodo;`

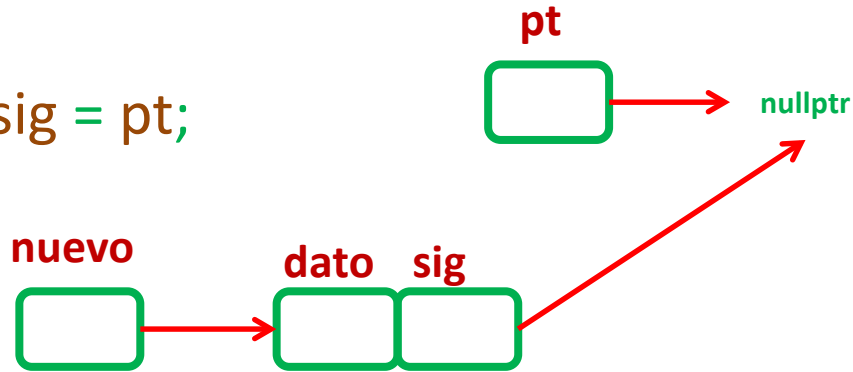


3. – `nuevo->dato = 5;`

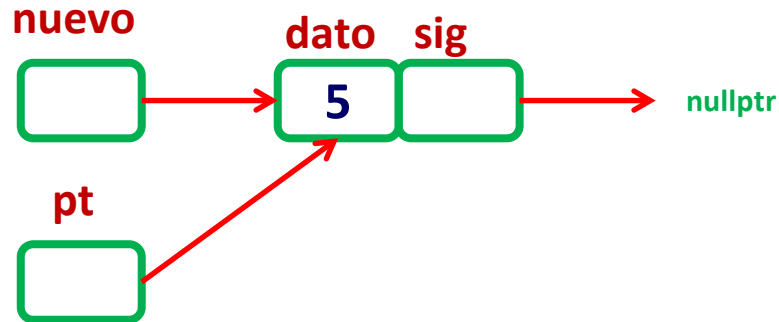


# Algoritmo para crear una lista ligada simple:

4. – `nuevo->sig = pt;`

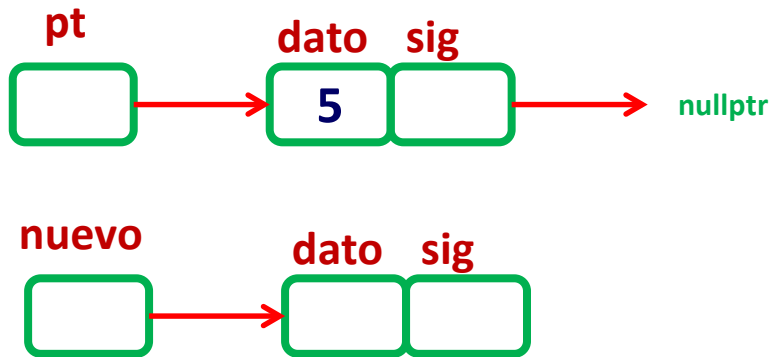


5. – `pt = nuevo;`



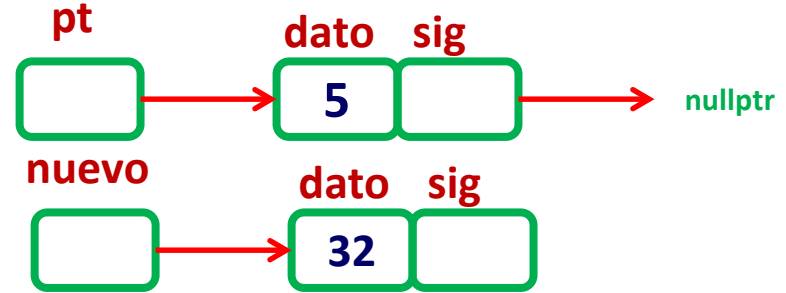
# Algoritmo para crear una lista ligada simple:

6. – `nuevo = new struct Nodo;`  
(repetimos el paso 2)

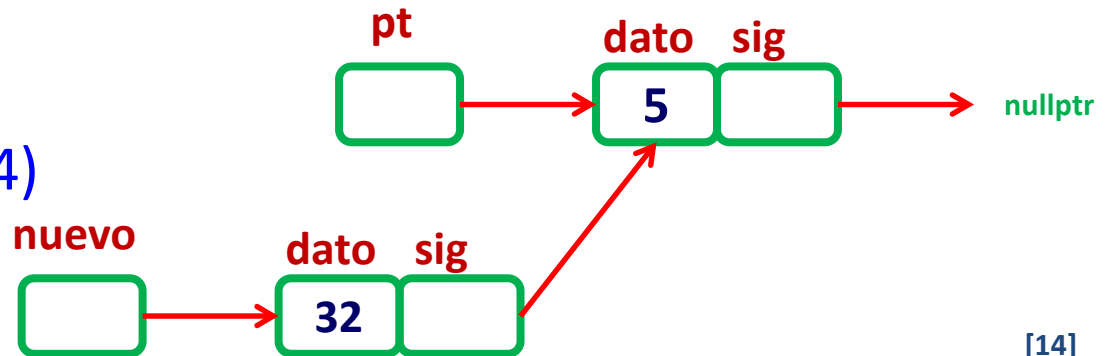


# Algoritmo para crear una lista ligada simple:

7. – nuevo->dato = 32;  
(repetimos el paso 3)

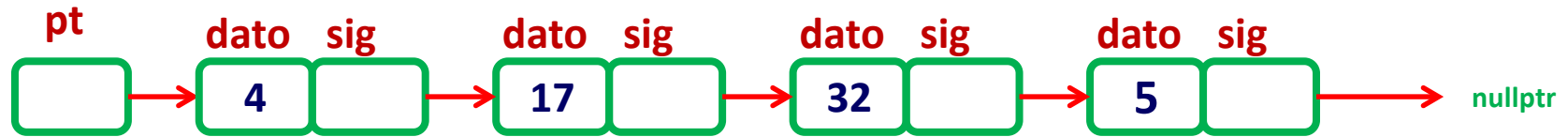


8. – nuevo->sig = pt;  
(repetimos el paso 4)

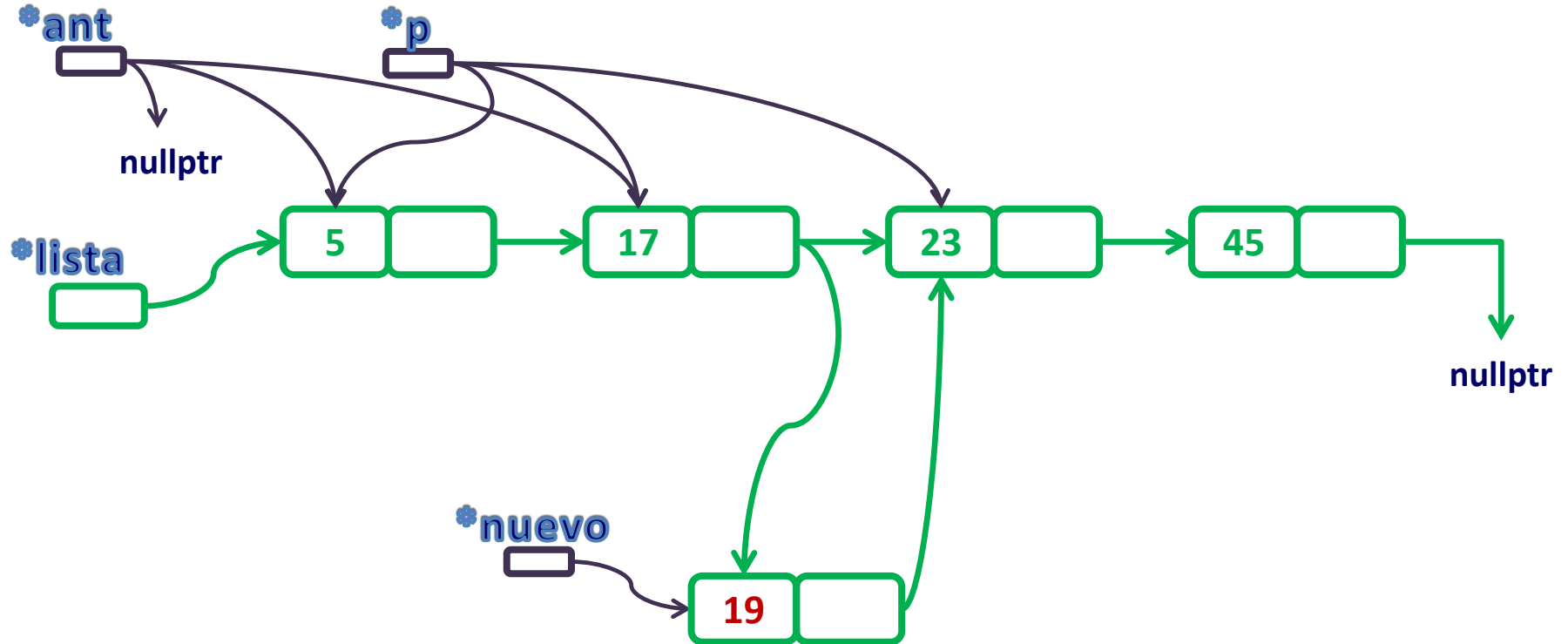


# Algoritmo para crear una lista ligada simple:

7. – Si repetimos los pasos 2, 3 y 4 varias veces



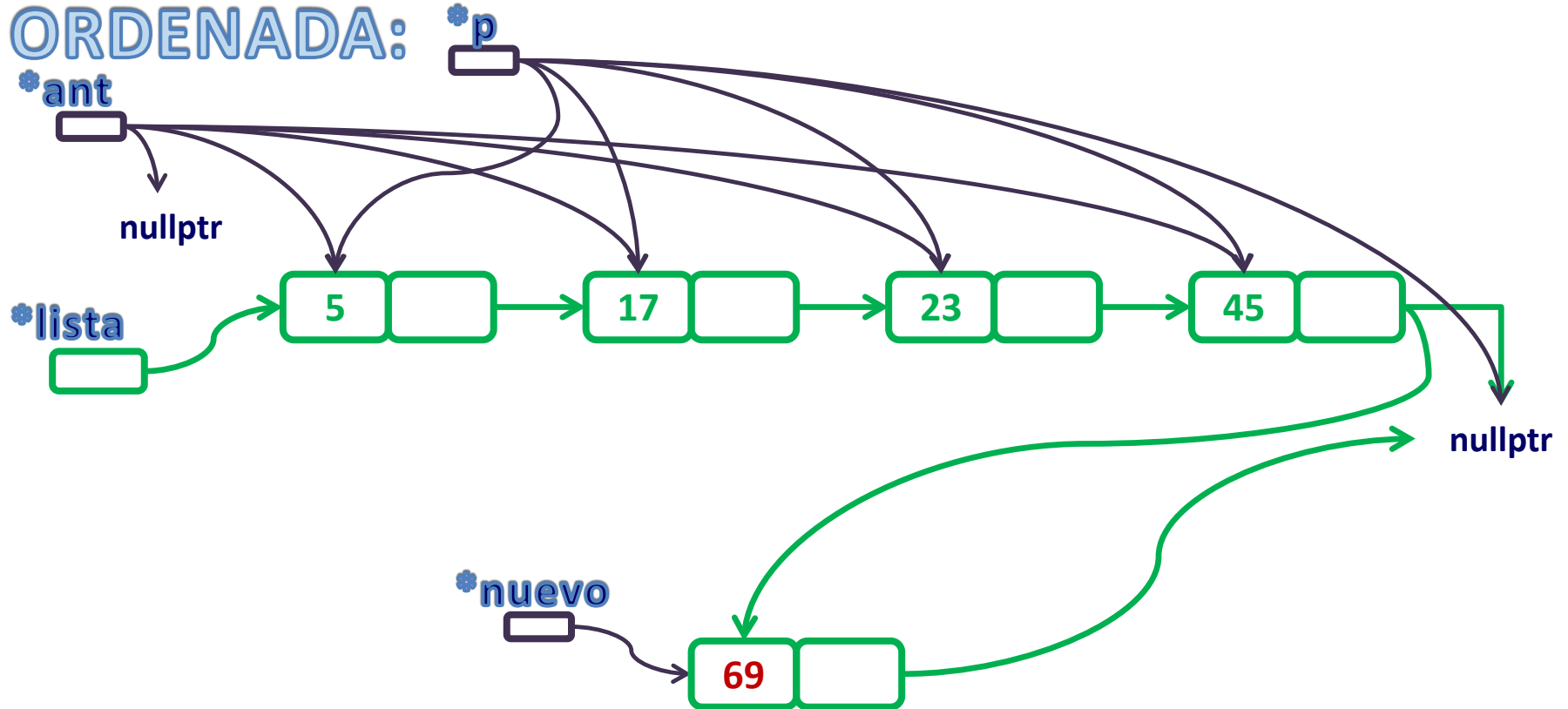
# INSERTAR UN DATO EN UNA LISTA LIGADA ORDENADA:





# INSERTAR UN DATO EN UNA LISTA LIGADA

ORDENADA:



# INSERTAR UN DATO EN UNA LISTA LIGADA

ORDENADA:

