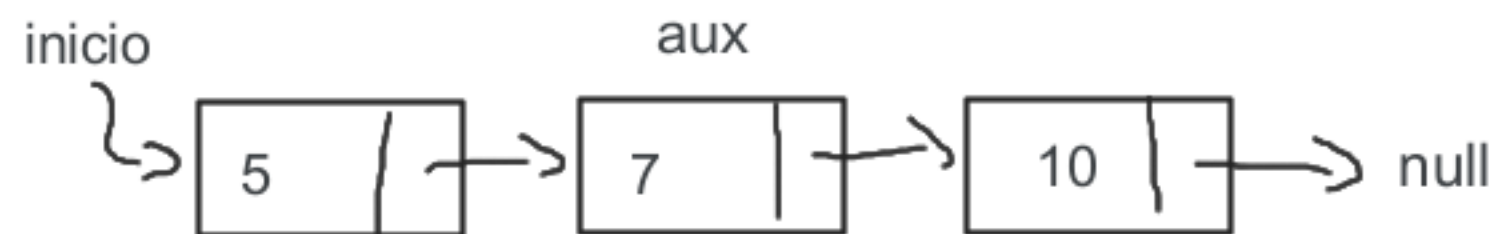


Introducción a la Programación y Computación 1 Sección E

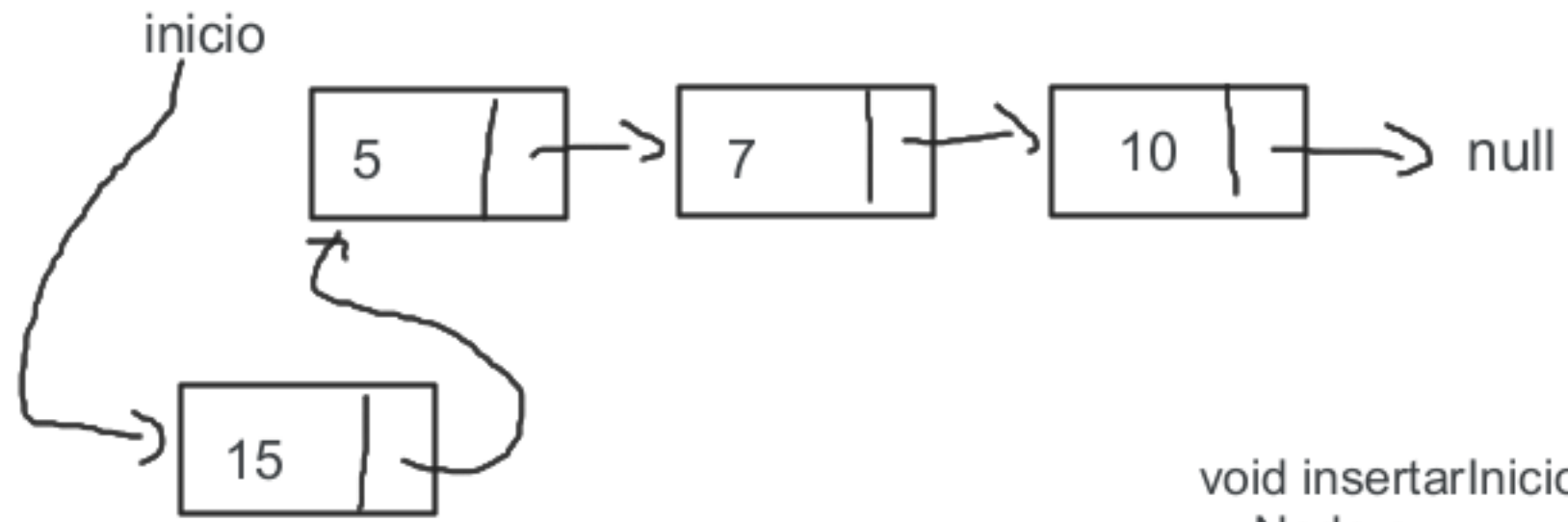
Ing. MSc. Neftalí Calderón

```
class Nodo {  
    int dato;  
    Nodo sig;  
  
    void Nodo(int n) {  
        dato = n;  
        sig = null;  
    }  
}
```

```
class Lista {  
    Nodo inicio;  
  
    void Lista() {  
        inicio = new Nodo();  
    }  
  
    void insertar() {  
    }  
  
    int extraer() {  
    }  
}
```

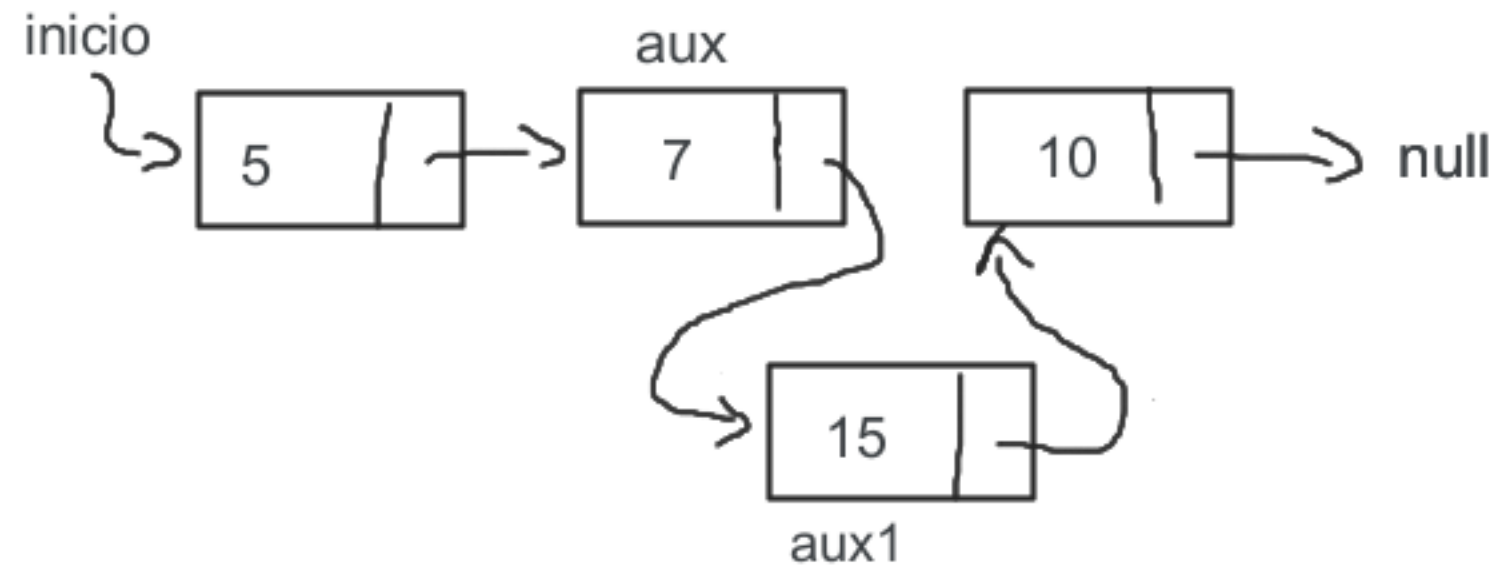


```
void insertarFinal(int n) {  
    Nodo aux;  
    aux = inicio;  
  
    if (inicio.sig == null) {  
        inicio.sig = new Nodo(n);  
    }  
    else {  
        while (aux.sig != null) {  
            aux = aux.sig;  
        }  
        aux.sig = new Nodo(n);  
    }  
}
```



```
void insertarInicio(int n) {  
    Nodo aux;  
  
    aux = new Nodo(n);  
    aux.sig = inicio.sig;  
    inicio.sig = aux;  
}
```

aux



```

void insertarMedio(int n, x) {
    Nodo aux;
    Nodo aux1;
    aux = inicio;

    while (aux.dato != x) {
        aux = aux.sig;
    }
    aux1 = new Nodo(n);
    aux1.sig = sig;
    aux.sig = aux1;
}

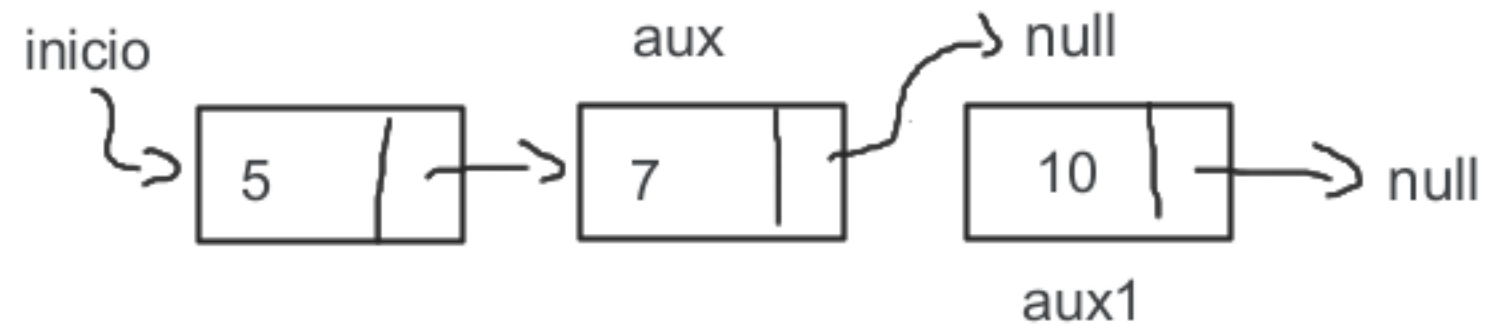
```

```

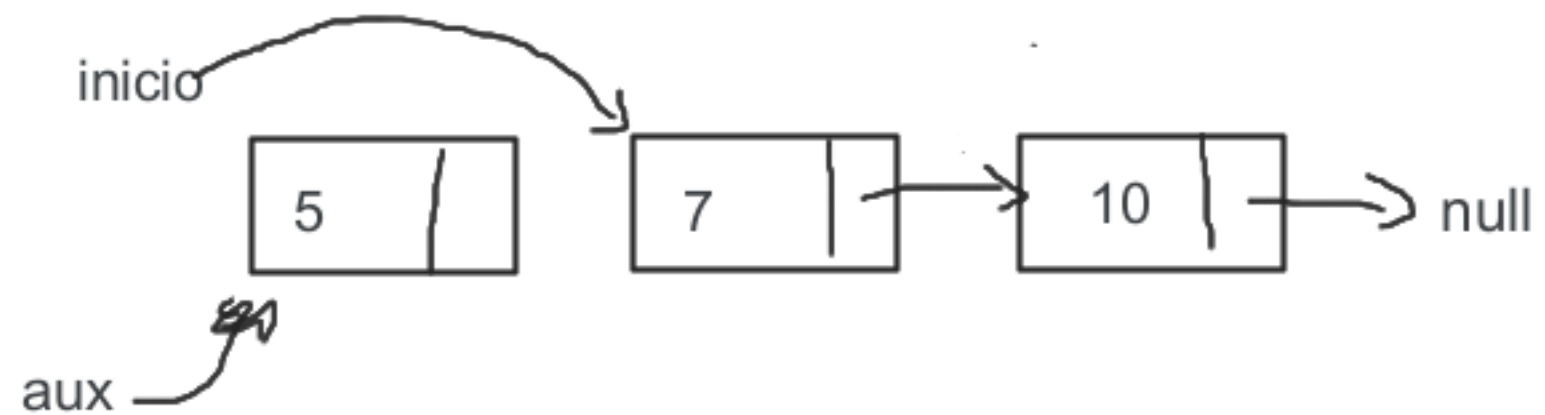
void insertarMedio(int n, x) {
    Nodo aux;
    Nodo aux1;
    aux = inicio;

    while (aux.sig.dato != x) {
        aux = aux.sig;
    }
    aux1 = new Nodo(n);
    aux1.sig = sig;
    aux.sig = aux1;
}

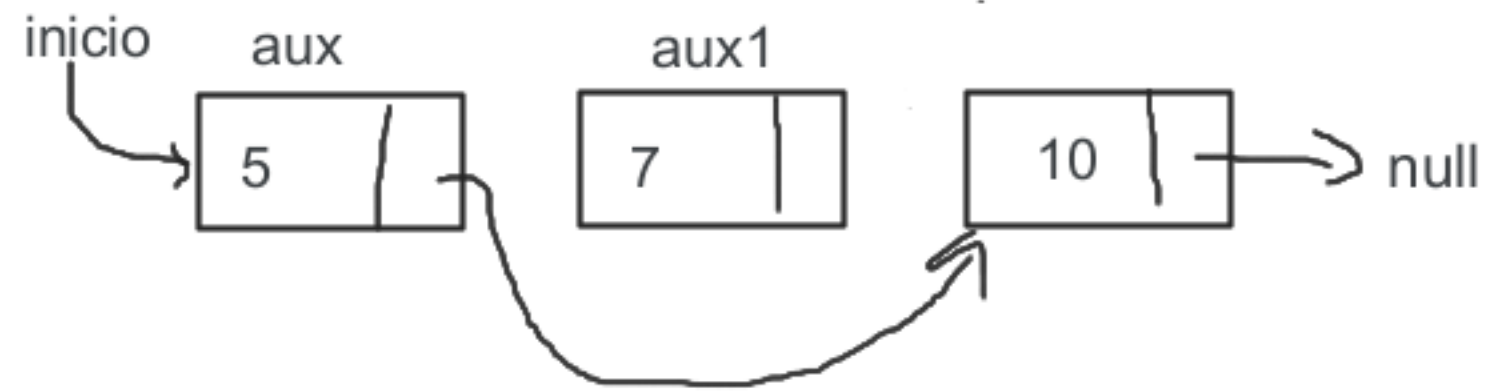
```



```
int eliminarUltimo() {  
    Nodo aux;  
    Nodo aux1;  
    aux = inicio;  
  
    if (inicio.sig == null) {  
        escribir("la lista está vacía");  
    }  
    while (aux.sig.sig != null) {  
        aux = aux.sig;  
    }  
    aux1 = aux.sig;  
    aux.sig = null;  
    return(aux1.dato);  
}
```



```
int eliminarPrimero() {  
    Nodo aux;  
    aux = inicio;  
  
    if (inicio.sig == null) {  
        escribir("la lista está vacía");  
    }  
    inicio = inicio.sig;  
    aux.sig = null;  
    return(aux.dato);  
}
```



```
int eliminarMedio(int x) {  
    Nodo aux;  
    Nodo aux1;  
    aux = inicio;  
  
    while (aux.sig.dato != x) {  
        aux = aux.sig;  
    }  
    aux1 = aux.sig;  
    aux.sig = aux1.sig;  
    aux1.sig = null;  
    return(aux1.dato);  
}
```



```
class Nodo {  
    int dato;  
    Nodo sig;  
    Nodo ant;  
  
    void Nodo(int n) {  
        dato = n;  
        sig = null;  
        ant = null;  
    }  
}
```

```
class Lista {  
    Nodo inicio;  
  
    void Lista() {  
        inicio = new Nodo();  
    }  
  
    void insertar() {  
    }  
  
    int extraer() {  
    }  
}
```



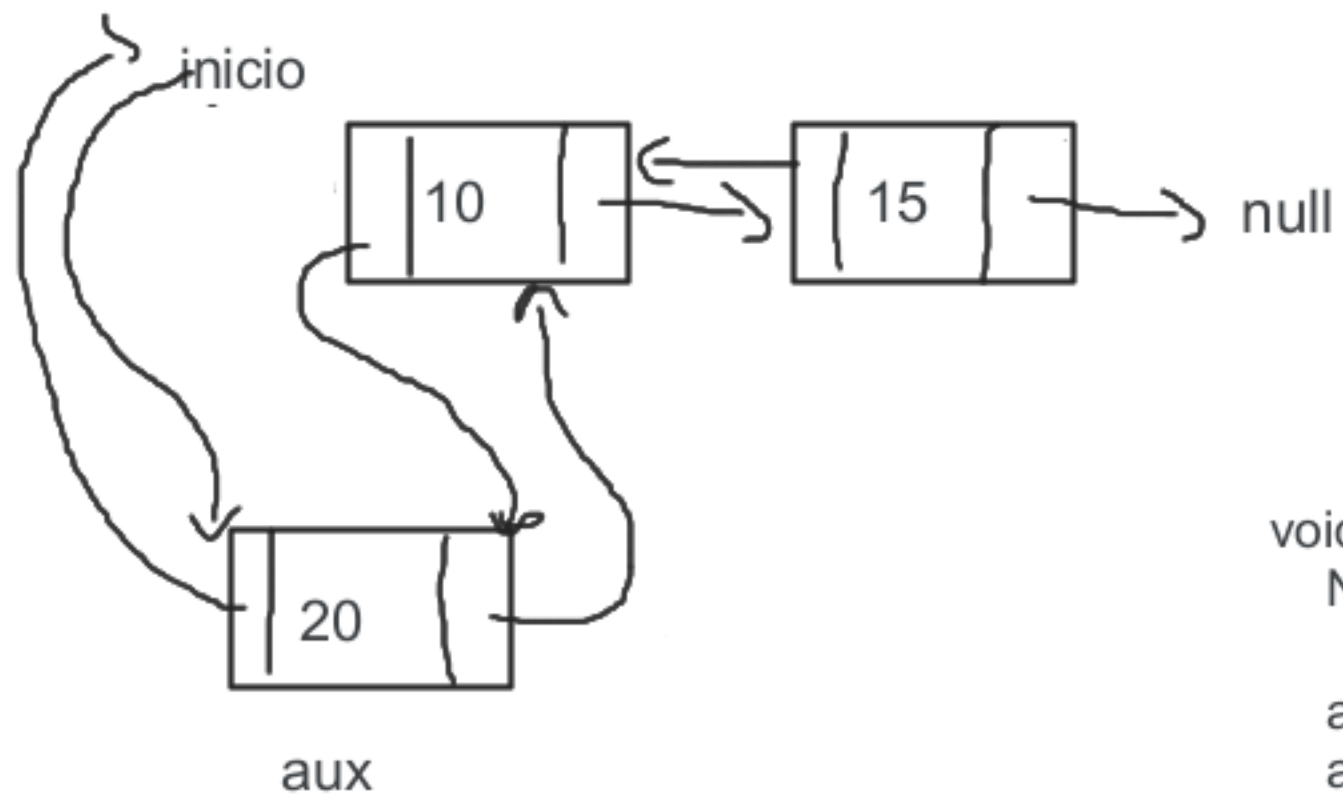


```

void insertarFinal(int n) {
    Nodo aux;
    aux = inicio;

    if (inicio.sig == null) {
        inicio.sig = new Nodo(n);
        inicio.sig.ant = inicio;
    }
    else {
        while (aux.sig != null) {
            aux = aux.sig;
        }
        aux.sig = new Nodo(n);
        aux.sig.ant = aux;
    }
}

```



```
void insertarInicio(int n) {  
    Nodo aux;  
  
    aux = new Nodo(n);  
    aux.sig = inicio.sig;  
    inicio.sig = aux;  
    aux.sig.ant = aux;  
    aux.ant = inicio;  
}
```



Estándares de convención de código

Conjunto de patrones, estilos y enfoques que nos ayudan a la mantenibilidad del código

java
css
WordPres

```
int eliminarUltimo()
{
    Nodo aux;
    Nodo aux1;
    aux = inicio;

    if (inicio.sig == null)
    {
        escribir("la lista está vacía");
    }
    while (aux.sig.sig != null)
    {
        aux = aux.sig;
    }
    aux1 = aux.sig;
    aux.sig = null;
    return(aux1.dato);
}
```

Lista
dato
IVA

if (variable1 > variable2) && (variable1 < variable3) ...

```
int eliminarUltimo() {
    Nodo aux;
    Nodo aux1;
    aux = inicio;

    if (inicio.sig == null) {
        escribir("la lista está vacía"); }
    while (aux.sig.sig != null) {
        aux = aux.sig; }
    aux1 = aux.sig;
    aux.sig = null;
    return(aux1.dato); }
```

camel case	pilaVacía	pilaLlena	esOperador	calculoArea
snake case	pila_vacia	pila_llena	es_operador	calculo_area
pascal case	PilaVaia	PilaLlena	EsOperador	CalculoArea

/*

autor:
descripcion:
versión:
fecha:

*/

fac_notasDeCredito
fac_calculoDeDescuentos
fac_promociones
inv_cardex
inv_ingresoMercaderia
inv_trasladoBodegas
mk_promociones

frm_factura
frm_notaCredito
rpt_ventasProducto
rpt_ventasVendedor
rpt_inventario