



Conceptos de Algoritmos Datos y Programas



CADP – TIPOS DE DATOS - LISTA



Creación de una lista.

Agregar nodos al comienzo de la lista.

Recorrido de una lista.

Agregar nodos al final de la lista.

Insertar nodos en una lista ordenada

Eliminar nodos de una lista



CADP – TEMAS



- Operación de AGREGAR ADELANTE
- Operación de AGREGAR AL FINAL

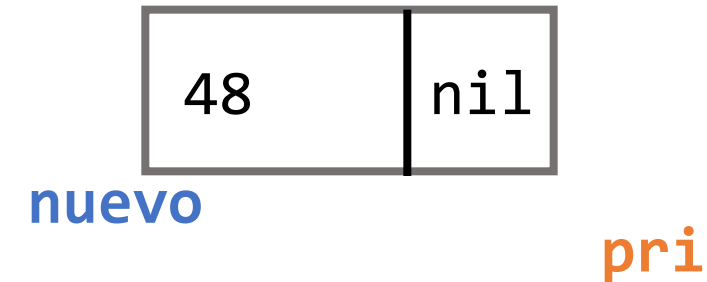
CADP – TIPOS DE DATOS - LISTA



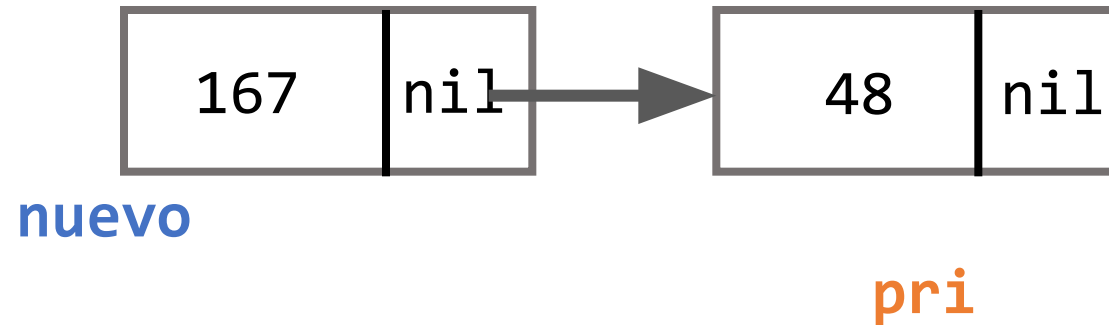
AGREGAR ADELANTE

Implica generar un nuevo nodo y agregarlo como primer elemento de la lista.

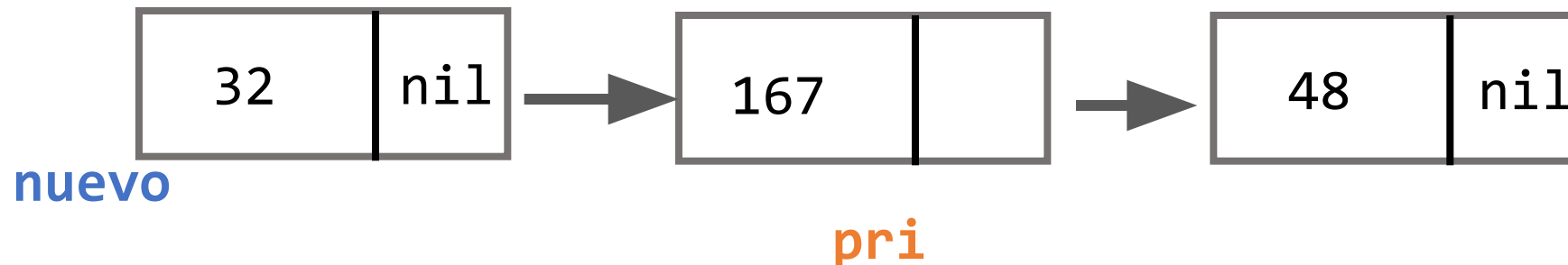
» `pri = nil`



» `pri <> nil`



Cómo lo escribo?





Implica generar un nuevo nodo y agregarlo como primer elemento de la lista.

Reservo espacio en memoria **nuevo elemento**.

si (es el primer elemento a agregar)
asigno al puntero inicial la dirección del **nuevo elemento**.

sino
indico que el siguiente de **nuevo elemento** es el puntero inicial.
actualizo el puntero inicial de la lista con la dirección del **nuevo elemento**.

CADP – TIPOS DE DATOS - LISTA

AGREGAR ADELANTE



```
Program uno;
```

```
Type listaE= ^datosEnteros;
```

```
    datosEnteros= record
                        elem:integer;
                        sig:listaE;
                    end;
```

```
Var
```

```
    pri: listaE;
    num:integer;
```

```
Begin
```

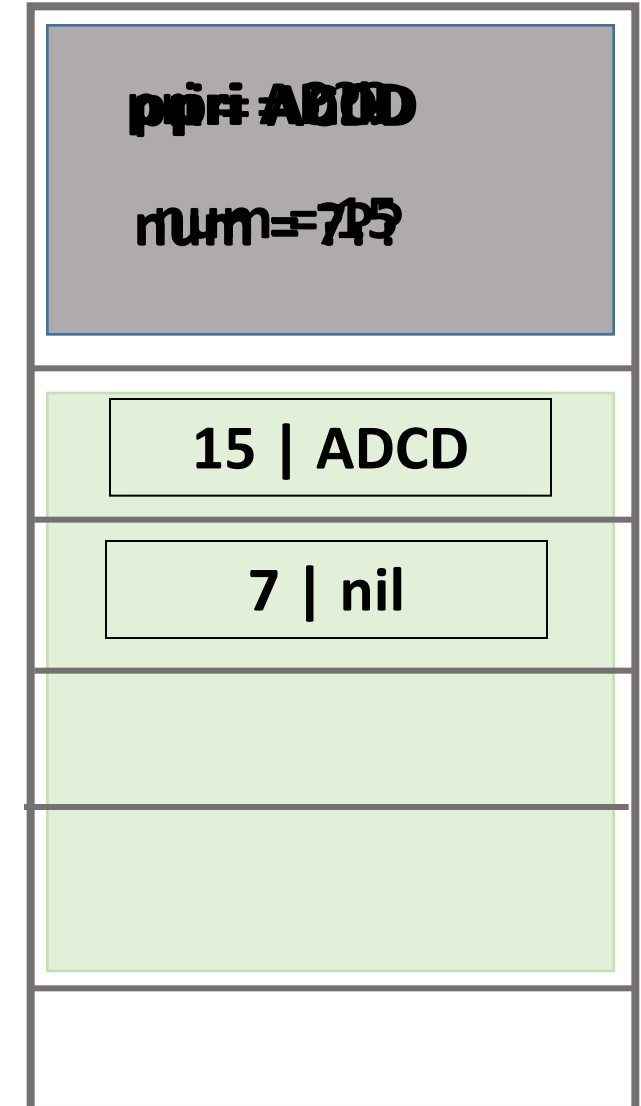
```
    crear (pri);
    read (num);
    agregarAdelante (pri,num);

    read (num);
    agregarAdelante (pri,num);
```

ACDD

ADCD

ADDA





```
procedure agregarAdelante (var pI:listaE; num:integer);
```

```
Var
```

```
nuevo:listaE;  creo espacio para el  
nuevo elemento
```

```
Begin
```

```
  new (nuevo); nuevo^.elem:= num; nuevo^.sig:=nil;
```

```
  if (pI = nil) then pI:= nuevo
```

```
  else begin
```

```
    nuevo^.sig:= pI;
```

```
    pI:=nuevo;
```

```
  end;
```

*Evalúo el caso y
reasigno los
punteros*

```
End;
```

CADP – TIPOS DE DATOS - LISTA



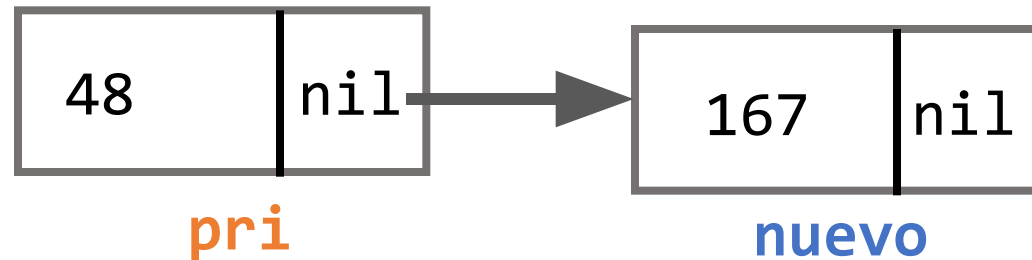
AGREGAR AL FINAL

Implica generar un nuevo nodo y agregarlo como último elemento de la lista.

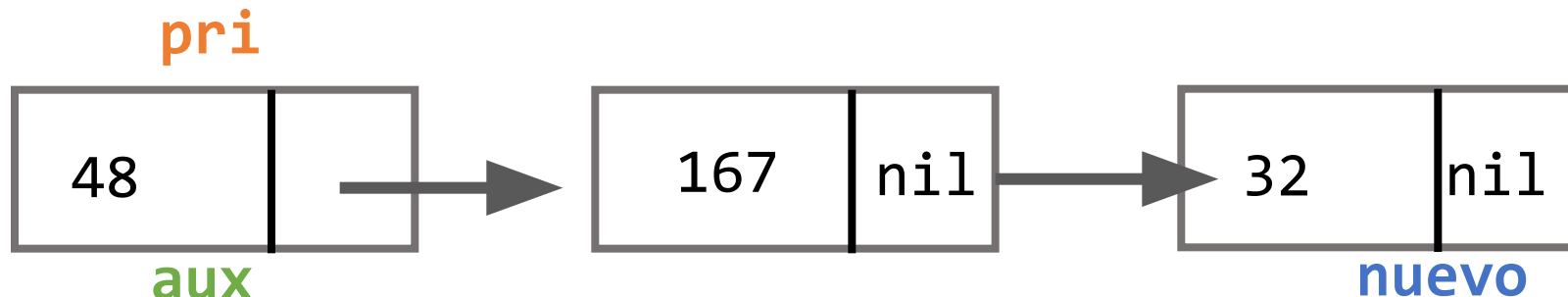
➤ `pri = nil`



➤ `pri <> nil`



Cómo lo escribo?





Implica generar un nuevo nodo y agregarlo como último elemento de la lista.

Reservo espacio en memoria **nuevo elemento**.

si (es el primer elemento a agregar)
 asigno al puntero inicial la dirección del **nuevo elemento**.

sino
 inicializo un puntero auxiliar **aux**
 mientras (no llegue al último elemento)
 avanzo en la lista.
 actualizo como siguiente del último nodo al **nuevo elemento**

CADP – TIPOS DE DATOS - LISTA

AGREGAR AL FINAL



```
Program uno;
```

```
Type listaE= ^datosEnteros;
```

```
    datosEnteros= record
                    elem:integer;
                    sig:listaE;
                end;
```

```
Var
```

```
    pri: listaE;
    num:integer;
```

```
Begin
```

```
    crear (pri);
    read (num);
    agregarAlFinal (pri,num);

    read (num);
    agregarAlFinal (pri,num);
```

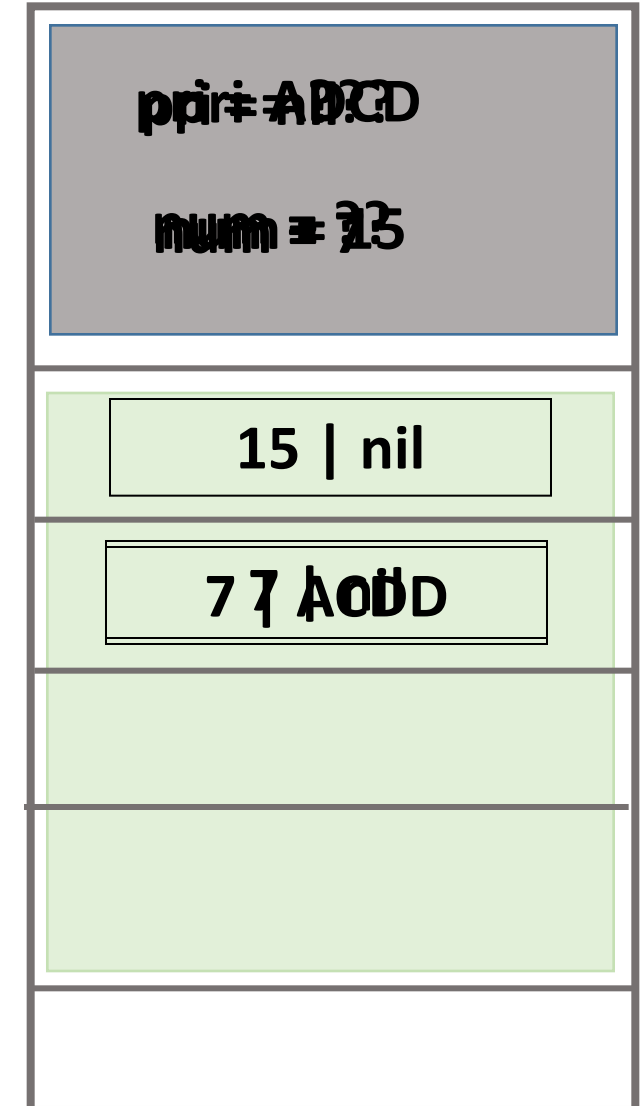
ACDD

15 | nil

ADCD

7 7 ADDD

ADDA



CADP – TIPOS DE DATOS - LISTA

AGREGAR AL FINAL



```
procedure agregarAlFinal (var pI:listaE; num:integer);
```

```
Var
```

```
    nuevo,aux:listaE;
```

```
Begin
```

```
    new (nuevo); nuevo^.elem:= num; nuevo^.sig:=nil;
```

```
    if (pI = nil) then pI:= nuevo  
    else begin
```

```
        aux:= pI;
```

```
        while (aux ^.sig <> nil) do
```

```
            aux:= aux^.sig;
```

```
        aux^.sig:=nuevo;  
    end;
```

```
End;
```

Si agrego al final por qué
paso por referencia el
puntero inicial?

Por qué en la
condición del while
se pregunta por el
aux^.sig?

← Evalúo si la lista está vacía

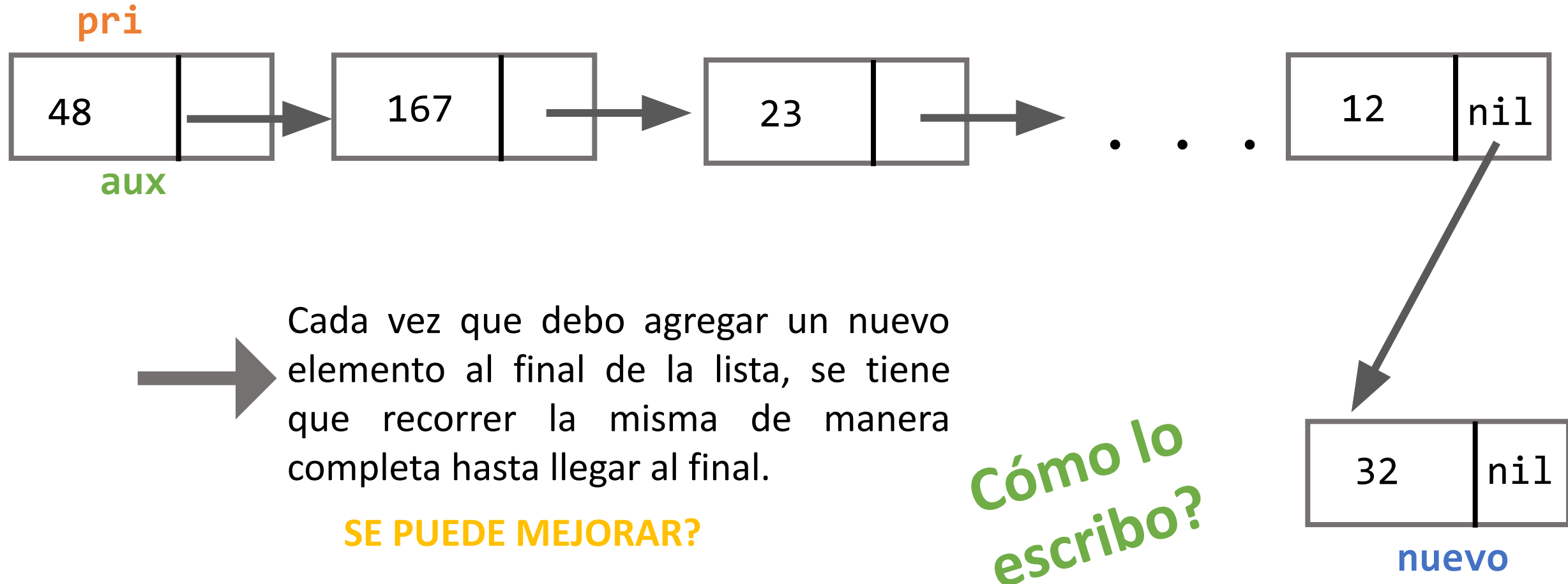
] Recorro y quedo
parado en el último
elemento

← Le indico al último que ahora
su siguiente es nuevo

CADP – TIPOS DE DATOS - LISTA

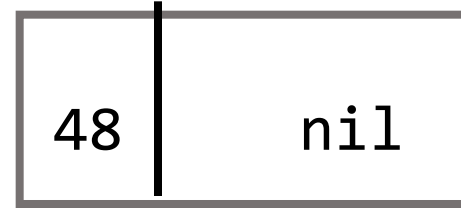


AGREGAR AL FINAL EN UNA LISTA (OPCION 2)





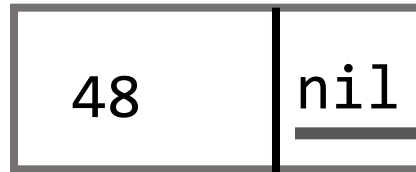
► `pri = nil`



nuevo

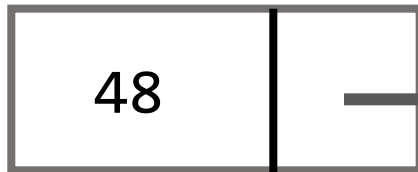
► `pri <> nil`

pri
ult

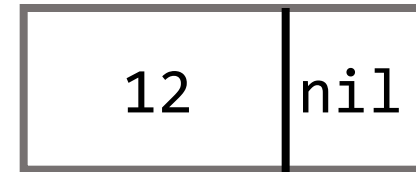
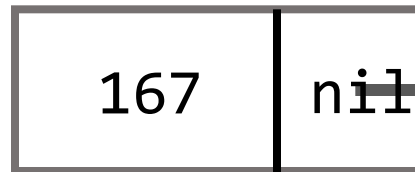


nuevo

pri



ult



nuevo

Cómo lo escribo?



AGREGAR AL FINAL EN UNA LISTA (OPCION 2)

Implica generar un nuevo nodo y agregarlo como último elemento de la lista.

Reservo espacio en memoria **nuevo elemento**.

si (es el primer elemento a agregar)

asigno al **puntero inicial** la dirección del **nuevo elemento**.

asigno al **puntero final** la dirección del **nuevo elemento**.

sino

actualizo como siguiente del **puntero final** al **nuevo elemento**

actualizo la dirección del **puntero final**

CADP – TIPOS DE DATOS - LISTA

AGREGAR AL FINAL -2



```
Program uno;
```

```
Type listaE= ^datosEnteros;
```

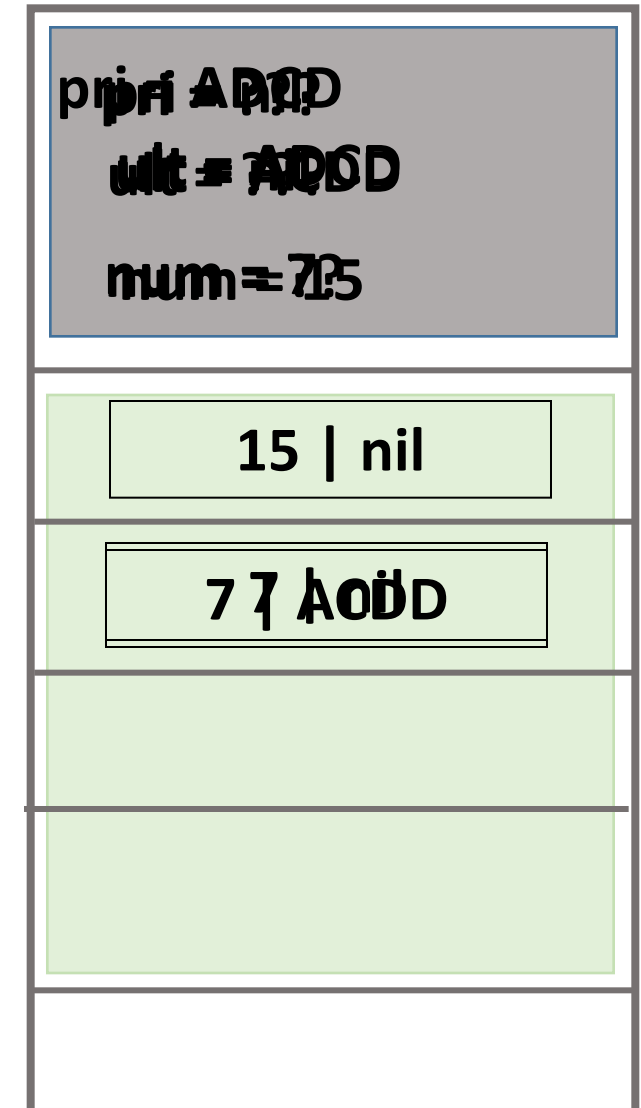
```
    datosEnteros= record
                        elem:integer;
                        sig:listaE;
                    end;
```

```
Var
```

```
    pri,ult: listaE;
    num:integer;
```

```
Begin
```

```
    crear (pri,ult);
    read (num);
    agregarAlFinal2 (pri,ult,num);
    read (num);
    agregarAlFinal2 (pri,,ult,num);
```





```
procedure agregarAlFinal2 (var pI,pU:listaE; num:integer);
```

```
Var
```

```
    nuevo:listaE;
```

```
Begin
```

```
    new (nuevo); nuevo^.elem:= num; nuevo^.sig:=nil;
```

```
    if (pI = nil) then begin
```

```
        pI:= nuevo;
```

```
        pU:= nuevo;
```

```
    end
```

```
    else begin
```

```
        pU^.sig:=nuevo;
```

```
        pU:= nuevo;
```

```
    end;
```

```
End;
```

← Evalúo si la lista está vacía

← Actualizo el siguiente del
último nodo y al último nodo