

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



- Estructura de Datos - LISTA

- Operación de CREACION

- Operación de RECORRIDO



CREAR UNA LISTA

Implica marcar que la lista no tiene una dirección inicial de comienzo.

```
Program uno;
```

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

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

```
Var
```

```
    pri: listaE; {Memoria estática reservada}
```

Qué valor se le asigna a un puntero para indicar que no tiene una dirección asignada?

CADP – TIPOS DE DATOS - LISTA

CREAR



```
Program uno;
```

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

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

```
Var  
    pri: listaE;
```

```
Begin  
    pri:=nil;  
End.
```

Por qué no se hace
new (pri)?

Se puede
modularizar el
crear?



CADP – TIPOS DE DATOS - LISTA

CREAR



```
Program uno;
```

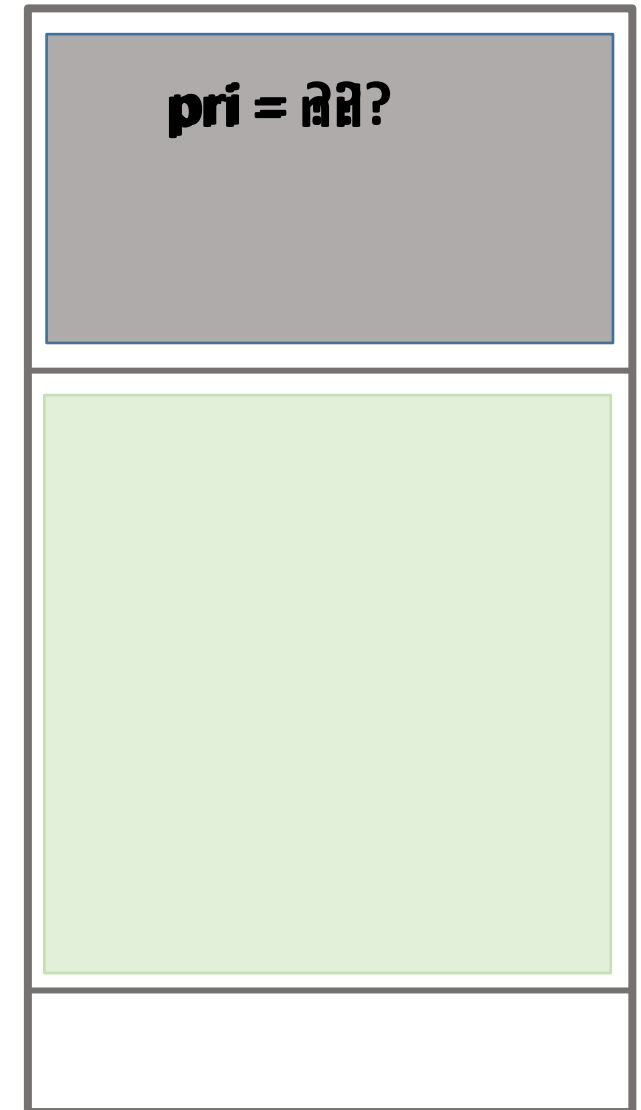
```
Type listaE= ^datosEnteros;
```

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

```
Procedure crear (var p: listaE);  
begin  
    p:= nil;  
end;
```

```
Var  
    pri: listaE;
```

```
Begin  
    crear (pri);  
End.
```





RECORRER UNA LISTA

Implica posicionarse al comienzo de la lista y a partir de allí ir “pasando” por cada elemento de la misma hasta llegar al final.

```
Program uno;
```

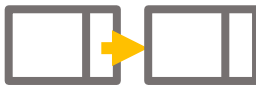
```
  Type listaE= ^datosEnteros;
```

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

```
  Var
    pri: listaE;
```

CADP – TIPOS DE DATOS - LISTA

RECORRER UNA LISTA



```
Program uno;
```

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

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

```
Var
```

```
    pri: listaE;
```

```
Begin
```

```
    crear (pri);
```

```
    cargarLista (pri); //Lo implementaremos más adelante
```

```
    recorrerLista (pri);
```

```
End.
```

ACDD

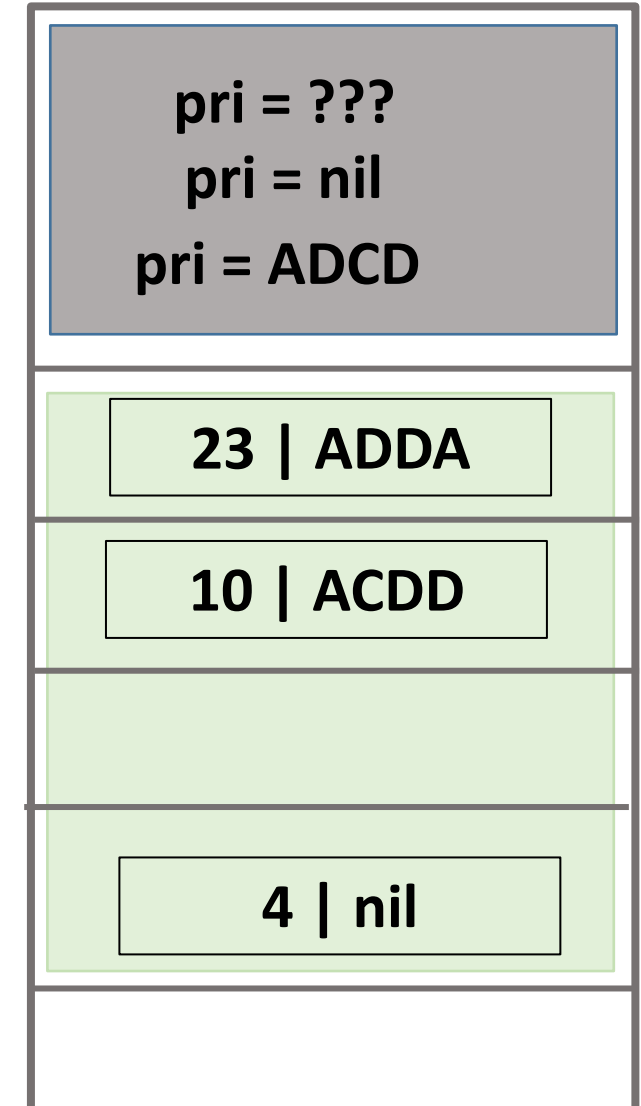
23 | ADDA

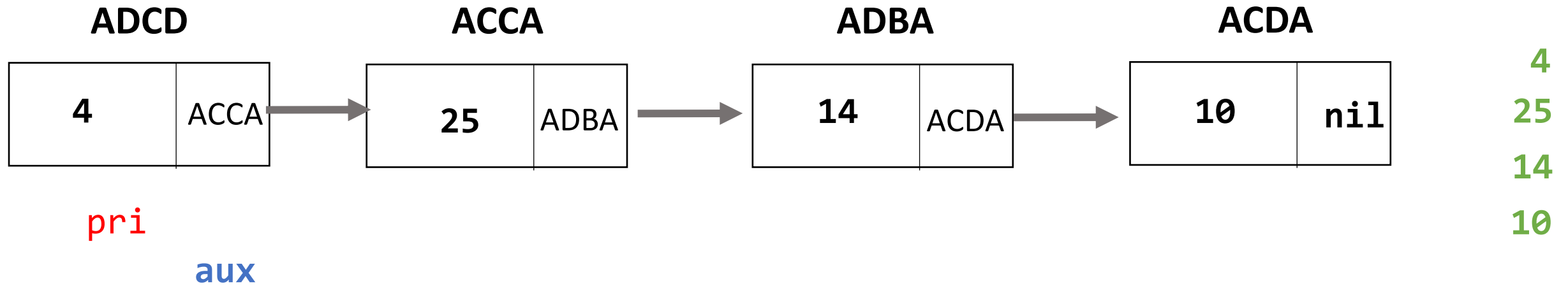
ADCD

10 | ACDD

ADDA

4 | nil



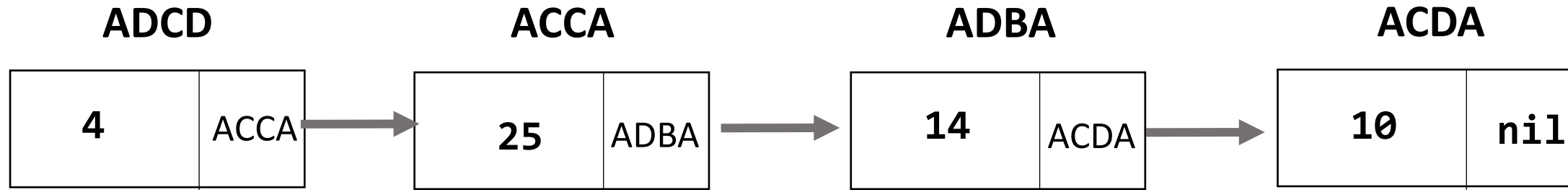


Inicializo una variable **auxiliar** con la dirección del puntero inicial de la lista

mientras (no sea el final de la lista)

 proceso el elemento (ej: imprimo, sumo, modifico)

 avanzo al siguiente elemento de **auxiliar**



pri

```
procedure recorrerLista (pI: listaE);
```

```
Var
```

```
    aux:listaE;
```

```
begin
```

```
    aux:= pI;
```

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

```
        begin
```

```
            write (aux^.elem);
```

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

```
        end;
```

```
end;
```

Es correcto?



Si la lista está **vacía**, (aux^.sig) da error.

Si la lista tiene **un solo elemento** (aux^.sig <> nil) da falso.

Si la lista tiene **muchos elementos** no imprime el último



```
procedure recorrerLista (pI: listaE);
```

```
Var
```

```
  aux:listaE;
```

Es necesaria
la variable
aux?

```
begin
```

```
  aux:= pI;
```

```
  while (aux <> nil) do
```

```
    begin
```

```
      write (aux^.elem);
```

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

```
    end;
```

```
end;
```

ALTERNATIVA

```
procedure recorrerLista (pI: listaE);
```

```
begin
```

```
  while (pI <> nil) do
```

```
    begin
```

```
      write (pI^.elem);
```

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

```
    end;
```

```
end;
```