

Lista dinámica circular. Inserción y eliminación.

```
function Crear_nodo (x:tinfo; var nuevo:tPos): boolean;
begin
  if Hay_memoria then
    begin
      new(nuevo);
      nuevo ^.info:=x;
      nuevo ^.sig:=nuevo; {Se apunta a si mismo}
      Crear_nodo:= true;
    end
  else Crear_nodo:=false;
end;
```

```
function Insertar (x:tinfo; P:tPos; var Lista:tLista):
  boolean;
  {PreCond: P posicion valida en la lista}
```

```
var aux: tPos ;
begin
  if Crear_nodo (x, nuevo)
  then begin
    Insertar:= true;
    if Es_vacia(Lista)
    then Lista := nuevo
    else begin
      Anterior(P, Lista) ^.sig:=nuevo;
      nuevo ^.sig:=P
    end
  end
  else Insertar:= false;
end;
```

```
function Anadir (x:tinfo; var Lista:tLista): boolean;
var nuevo: tPos ;
begin
  if Crear_nodo (x, nuevo)
  then begin
    if Es_vacia(Lista)
    then Lista := nuevo
    else begin
      nuevo ^.sig:= Lista ^.sig;
      Lista ^.sig:= nuevo;
      Lista := nuevo
    end;
    Anadir:= true
  end
  else Anadir:= false;
end;
```

```

function Eliminar_contenido(x:tInfo; var Lista:tLista)
  : boolean;
var ant,p:tpos;
begin
  if Es_vacia(Lista)
  then Eliminar_contenido:=false
  else begin
    p:=lista;
    if p^.info=x
    then begin
      if p=p^.sig
      {si solo hay un elemento}
      then lista:=nulo
      else begin
        lista:=anterior(p, Lista);
        lista^.sig:=p^.sig;
      end
      dispose(p);
      Eliminar_contenido:=true;
    end
  else begin
    {Busca el elemento a borrar. Guarda la
     posicion del anterior}

    while (p^.info <>x) and (p^.sig <> lista) do
    begin
      ant:=p;
      p:=p^.sig;
    end;
    if p^.info <>x
    then Eliminar_contenido:=false
    else begin
      ant^.sig:=p^.sig;
      dispose(P);
      Eliminar_contenido:=true;
    end
  end
end;

```