

# Tipos de Datos Abstractos

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

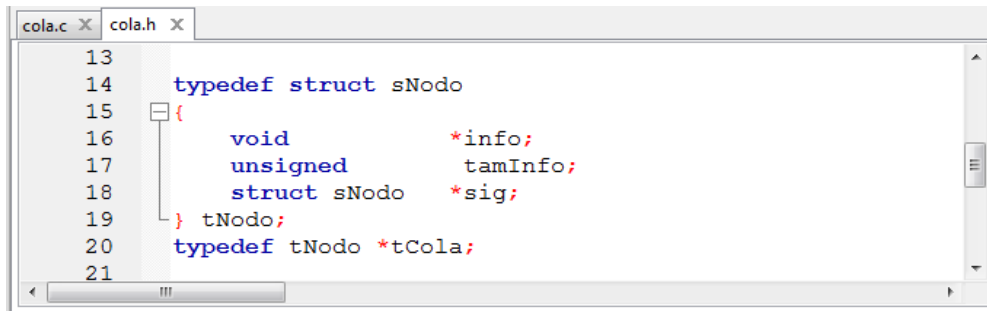
# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

tCola

info tam sig  
tNode



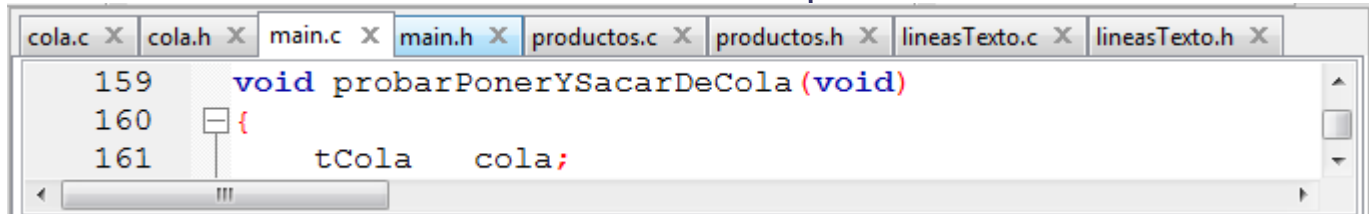
```
13
14 typedef struct sNode
15 {
16     void *info;
17     unsigned tamInfo;
18     struct sNode *sig;
19 } tNode;
20 typedef tNode *tCola;
21
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

?  
cola

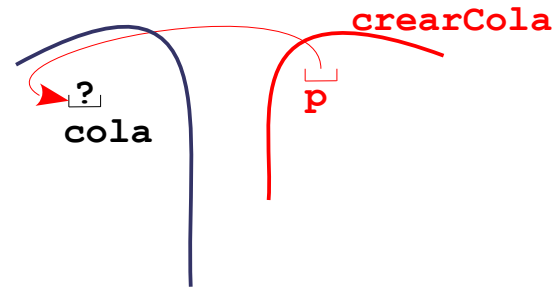


```
cola.c X cola.h X main.c X main.h X productos.c X productos.h X lineasTexto.c X lineasTexto.h X
159 void probarPonerYSacarDeCola(void)
160 {
161     tCola cola;
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

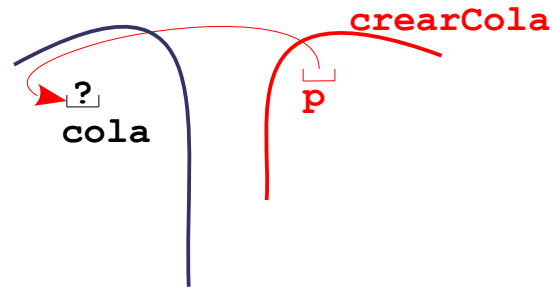


```
cola.c X cola.h X main.c X main.h X productos.c X productos.h X lineasTexto.c X lineasTexto.h X
159 void probarPonerYSacarDeCola(void)
160 {
161     tCola cola;
162     int cant;
163
164     crearCola(&cola);
165 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

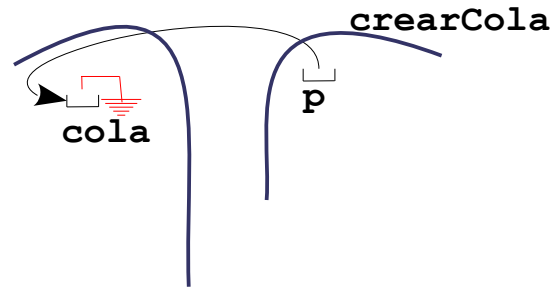


```
cola.c X cola.h X main.c X main.h X productos.c X productos.h X lineasTexto.c X lineasTexto.h X
159 void probarPonerYSacarDeCola(void)
160 {
161     tCola cola;
162     int cant;
163
164     crearCola
165
cola.c X cola.h X *main.c X main.h X productos.c X productos.h X
8 void crearCola(tCola *p)
9 {
10     *p = NULL;
11 }
12
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c X cola.h X main.c X main.h X productos.c X productos.h X lineasTexto.c X lineasTexto.h X
159 void probarPonerYSacarDeCola(void)
160 {
161     tCola cola;
162     int cant;
163
164     crearCo
165
cola.c X cola.h X *main.c X main.h X productos.c X productos.h X
8 void crearCola(tCola *p)
9 {
10     *p = NULL;
11 }
12
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



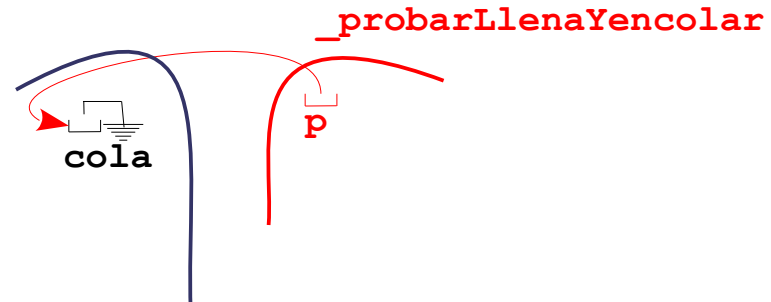
```
cola.c X cola.h X main.c X main.h X productos.c X productos.h X lineasTexto.c X lineasTexto.h X
159 void probarPonerYSacarDeCola(void)
160 {
161     tCola cola;
162     int cant;
163
164     crearCola(&cola);
165 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

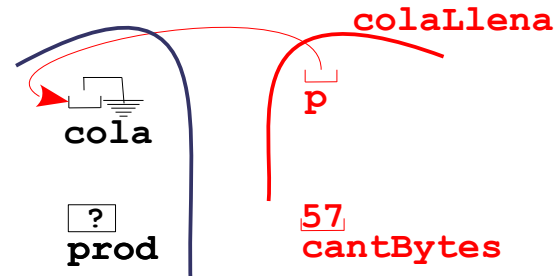


```
cola.c X cola.h X *main.c X main.h X
161     tCola  cola;
162     int    cant;
163
164     crearCola(&cola);
165
166     puts("Probando primitivas de cola con r
167         "===== ===== == == =
168     cant = probarLlenaYencolar(&cola);
169     printf("se pusieron %d productos en la
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

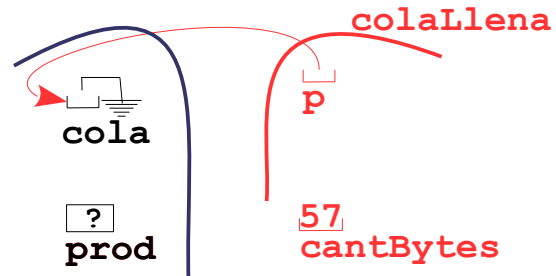


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



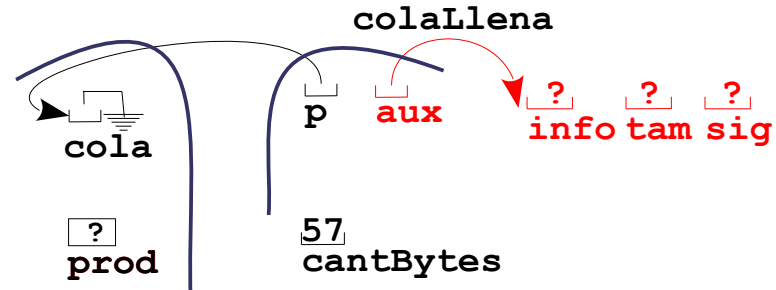
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



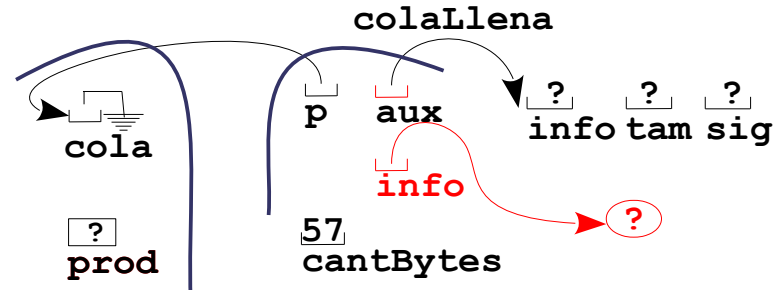
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNode *aux = (tNode *)malloc(sizeof(tNode));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c X cola.h X *main.c X main.h X
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

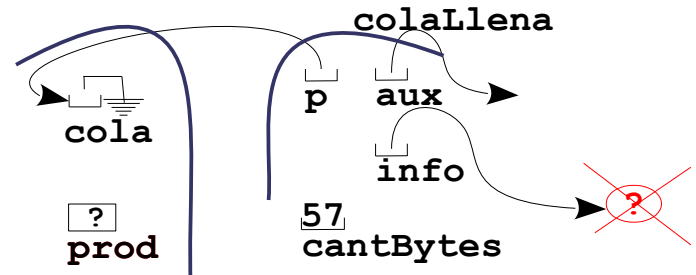
cola.c X cola.h X *main.c X main.h X
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



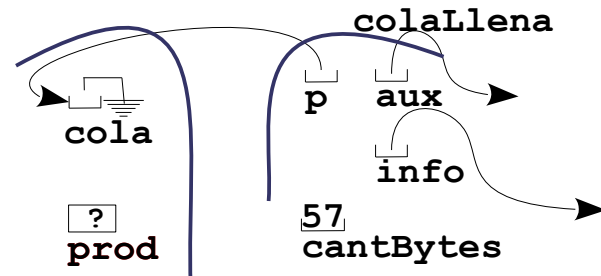
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```



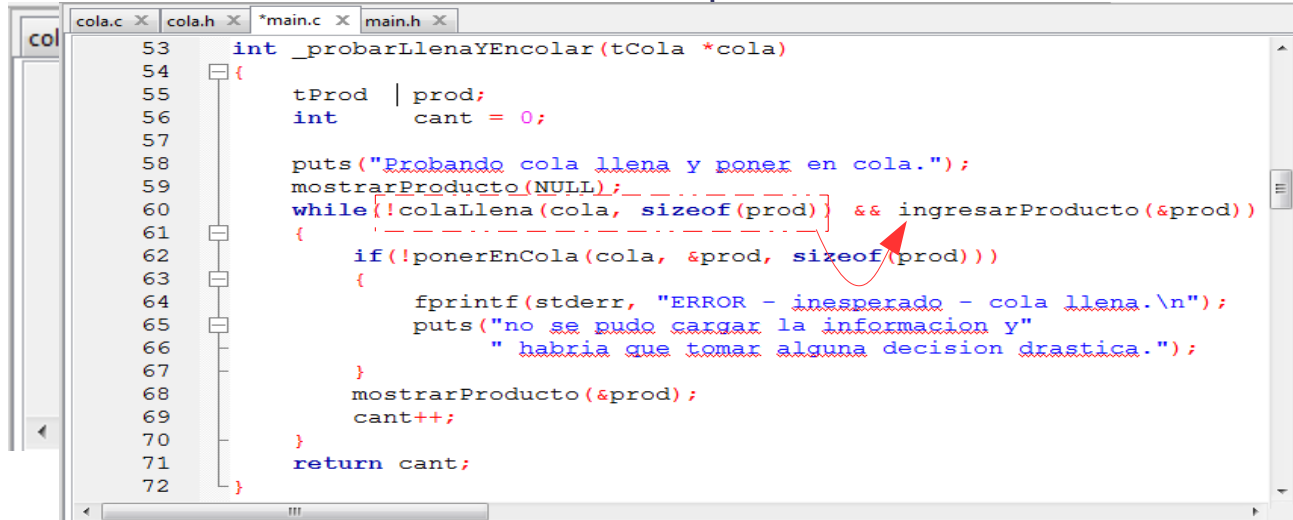
# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

cola

?  
prod

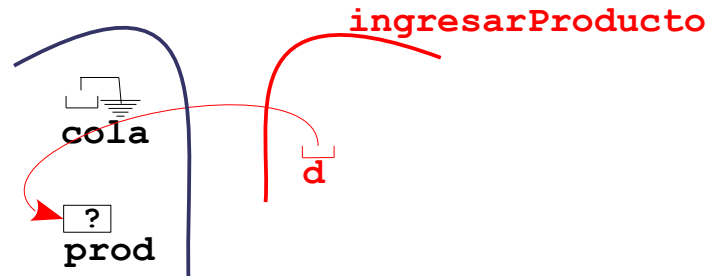


```
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la information y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

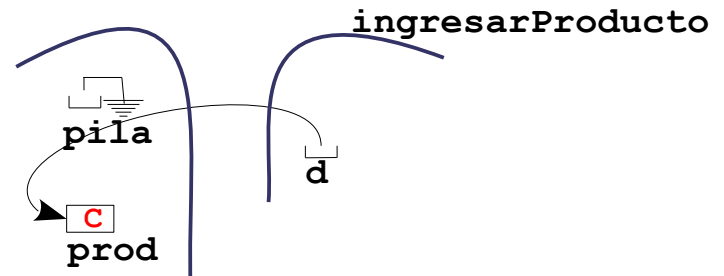


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

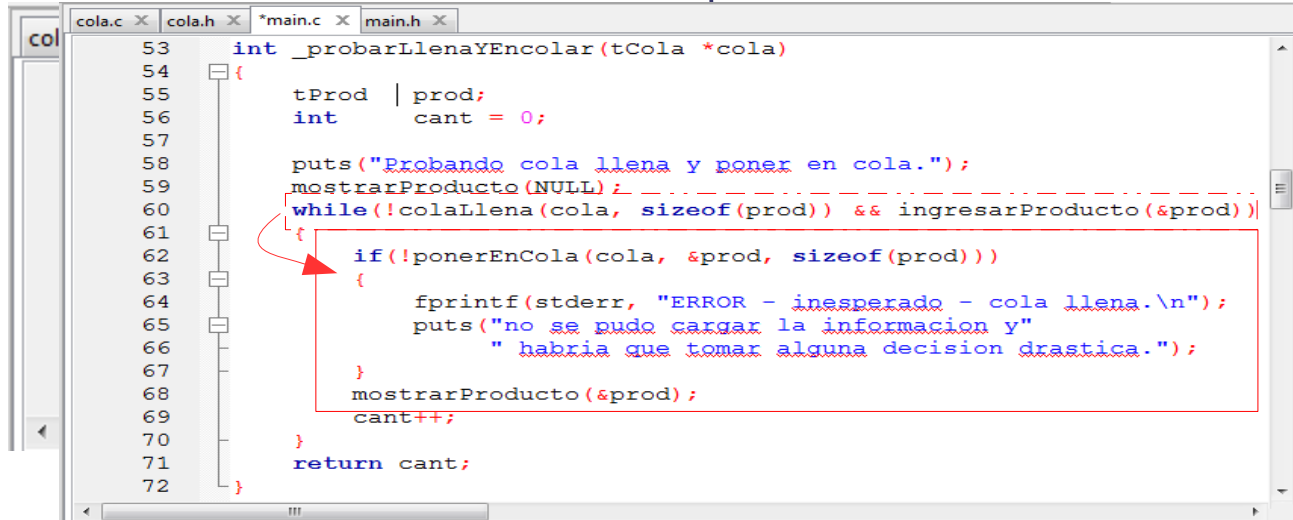
# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

cola

C  
prod

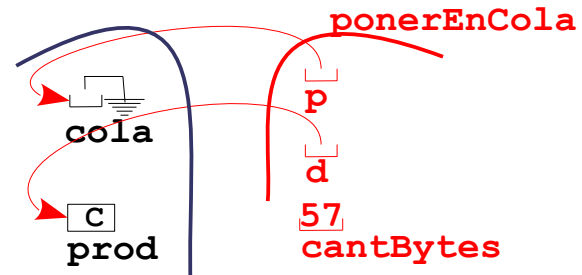


```
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la information y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

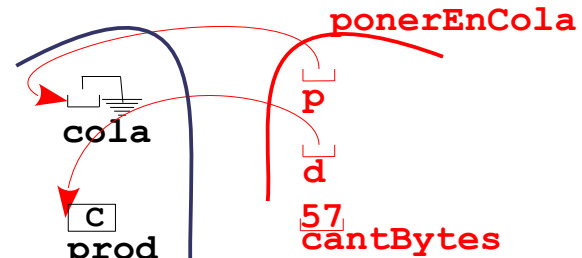


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

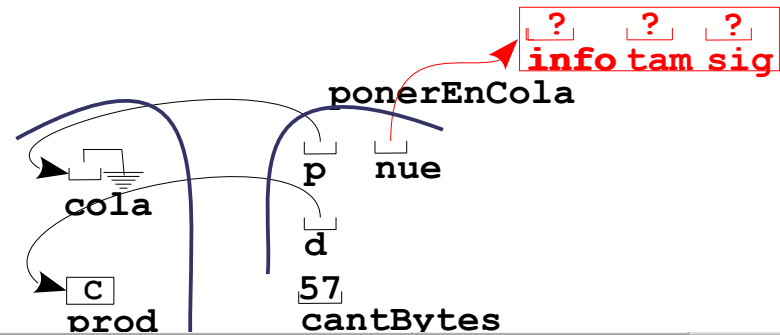


```
cola.c X cola.h X *main.c X main.h X
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

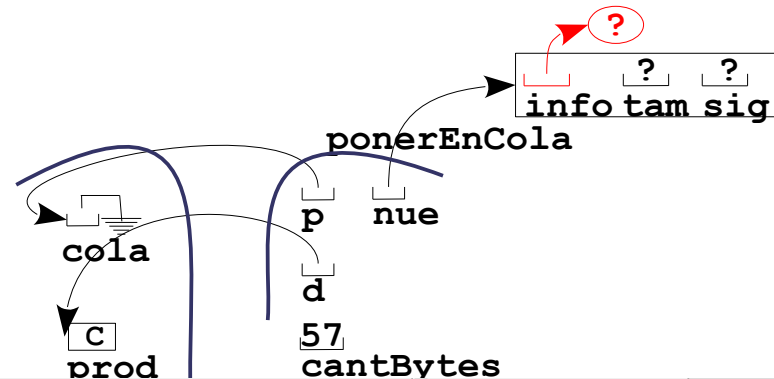


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
27        (nue->info = malloc(cantBytes)) == NULL)
28     {
29         free(nue);
30         return 0;
31     }
32     memcpy(nue->info, d, cantBytes);
33     nue->tamInfo = cantBytes;
34     if(*p == NULL)
35         nue->sig = nue;
36     else
37     {
38         nue->sig = (*p)->sig;
39         (*p)->sig = nue;
40     }
41     *p = nue;
42     return 1;
43 }
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



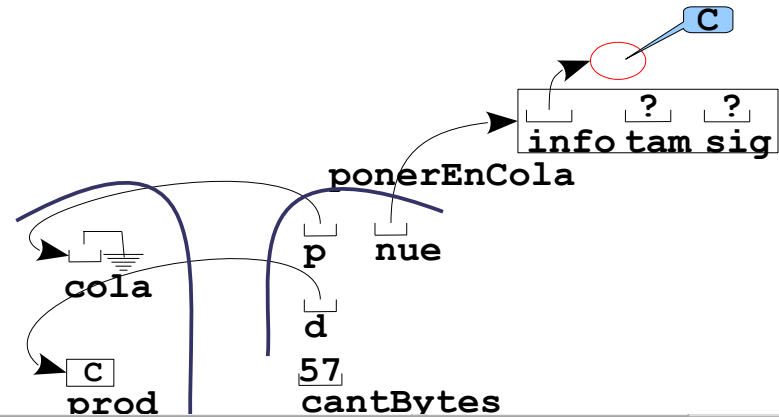
```
cola.c X cola.h X *main.c X main.h X
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
27        (nue->info = malloc(cantBytes)) == NULL)
28     {
29         free(nue);
30         return 0;
31     }
32     memcpy(nue->info, d, cantBytes);
33     nue->tamInfo = cantBytes;
34     if(*p == NULL)
35         nue->sig = nue;
36     else
37     {
38         nue->sig = (*p)->sig;
39         (*p)->sig = nue;
40     }
41     *p = nue;
42     return 1;
43 }
44 }
```



# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

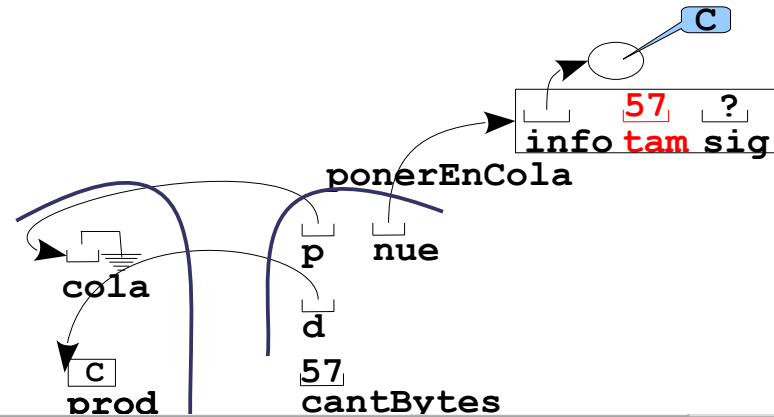


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

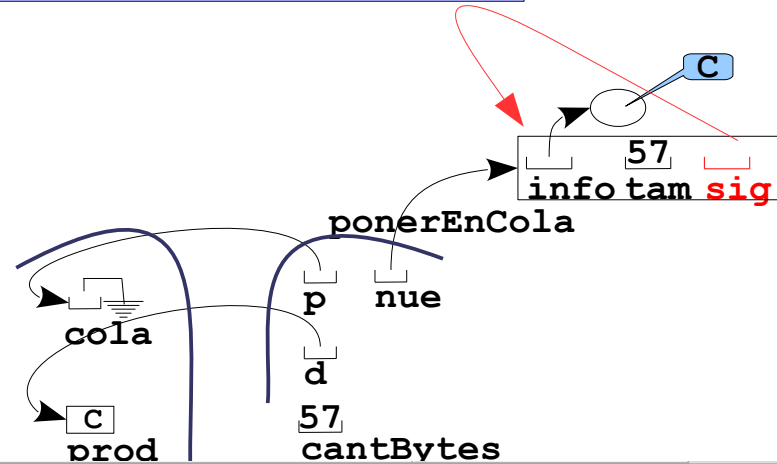


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

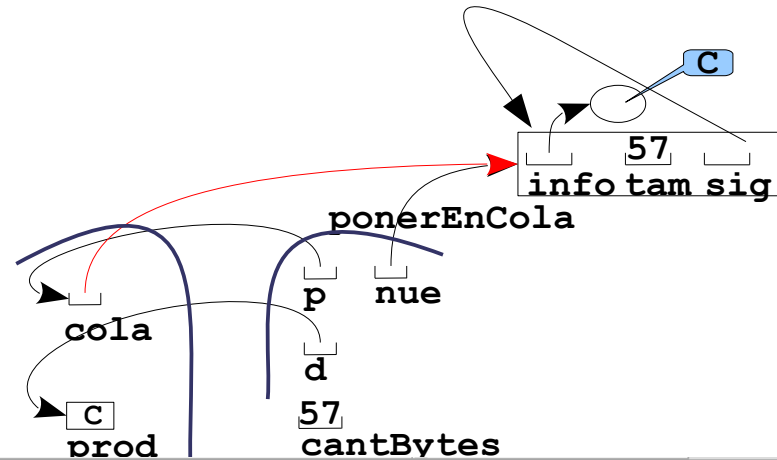


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36     {
37         nue->sig = nue;
38     }
39     else
40     {
41         nue->sig = (*p)->sig;
42         (*p)->sig = nue;
43     }
44     *p = nue;
45     return 1;
46 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

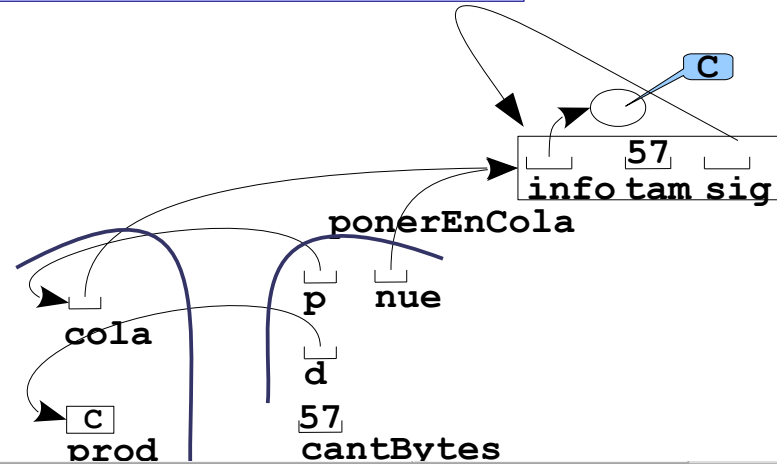


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

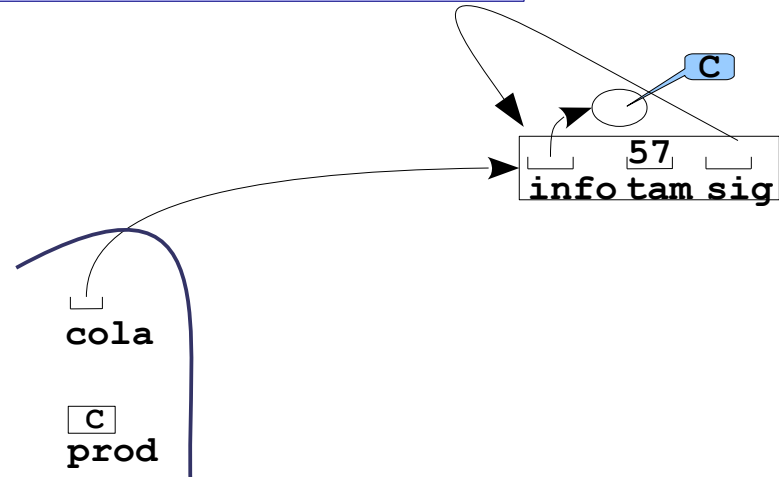


```
cola.c X cola.h X *main.c X main.h X
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

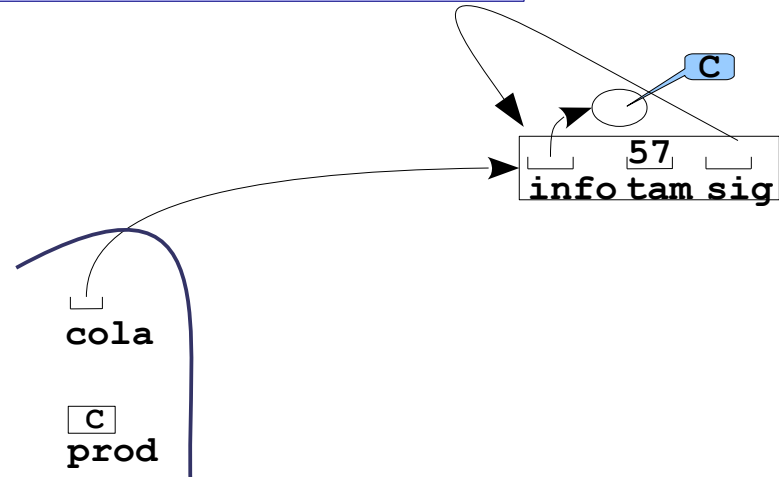


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

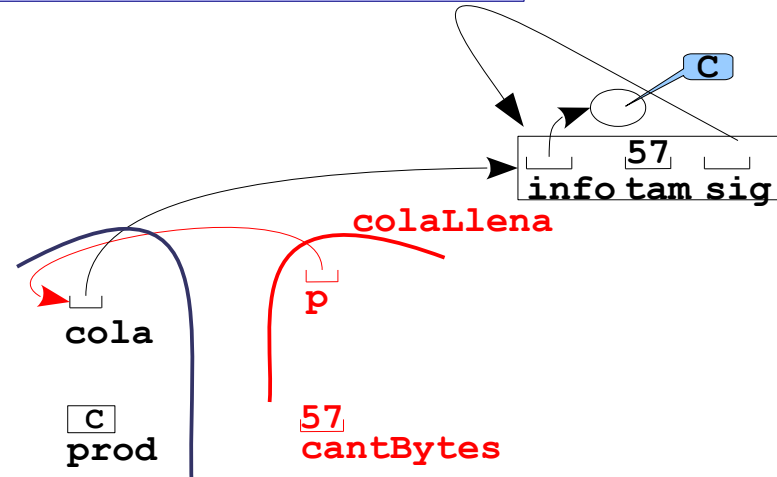


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la information y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



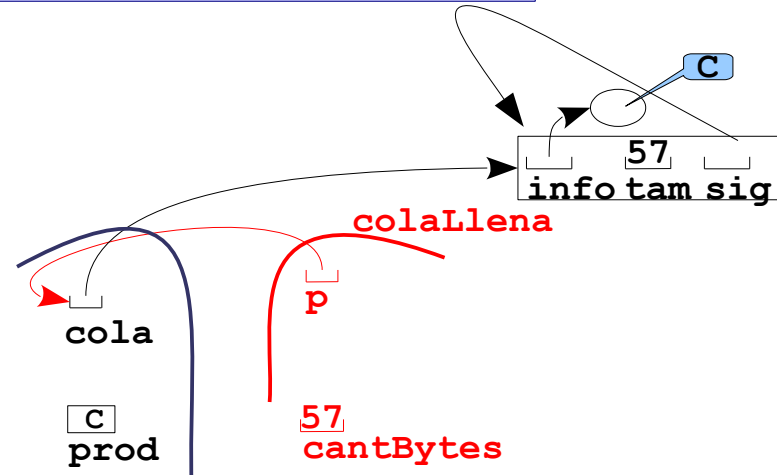
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



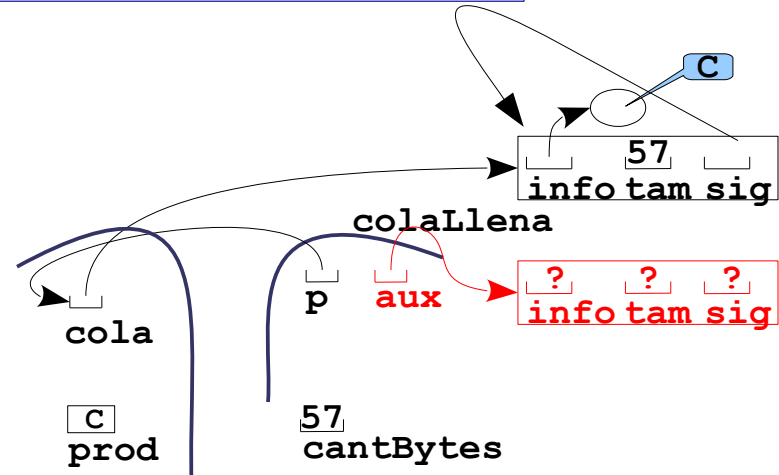
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



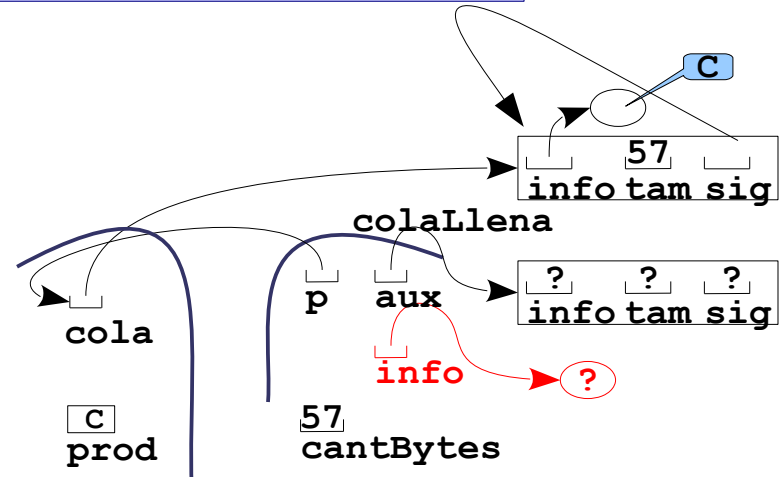
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



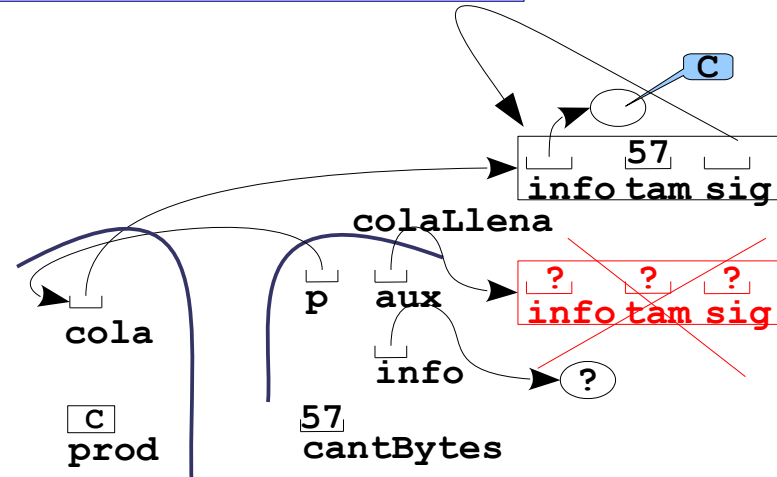
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



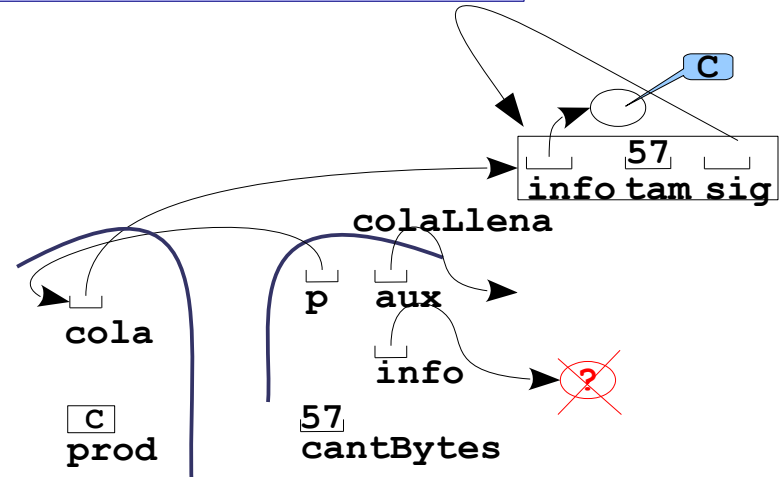
```
cola.c X cola.h X *main.c X main.h X
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c X cola.h X *main.c X main.h X
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



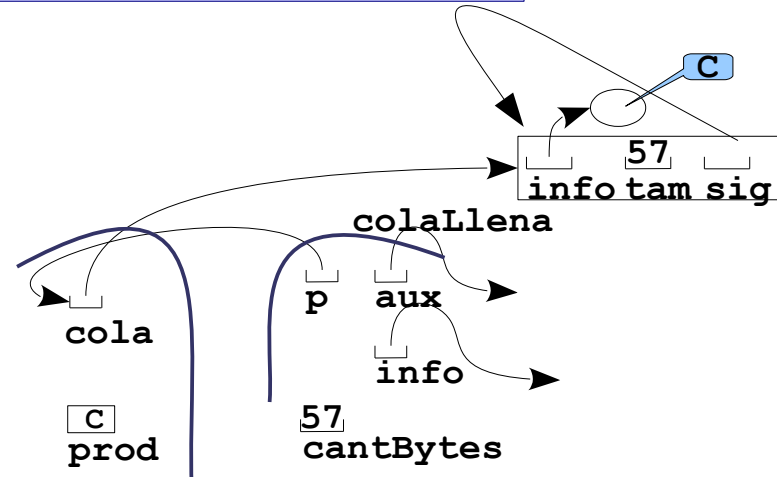
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



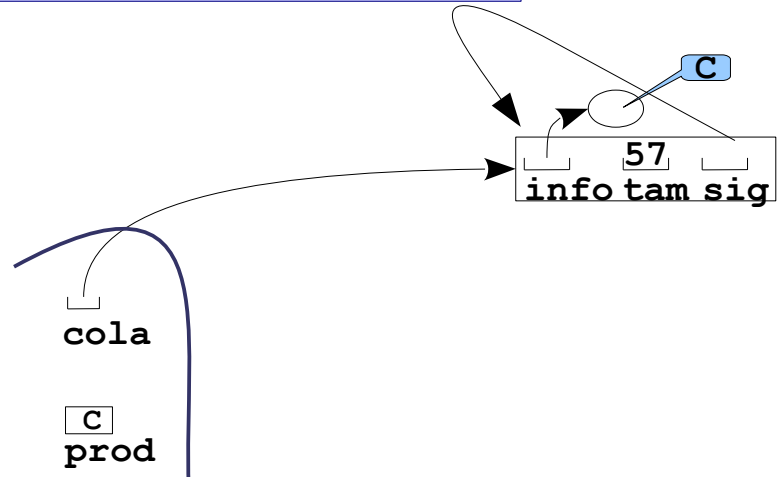
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

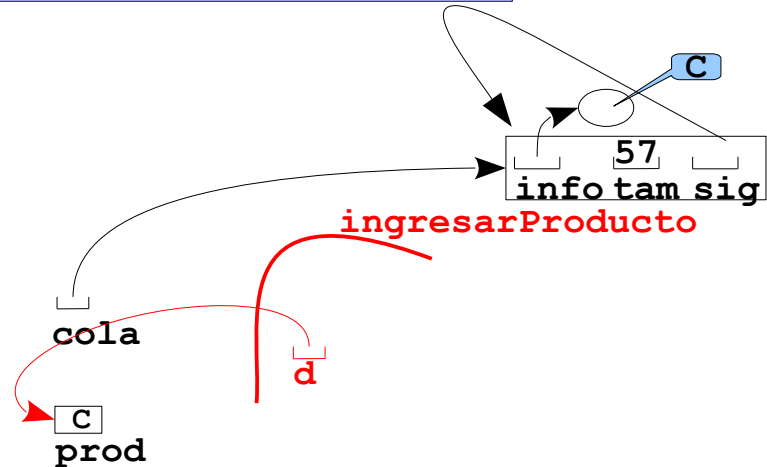


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



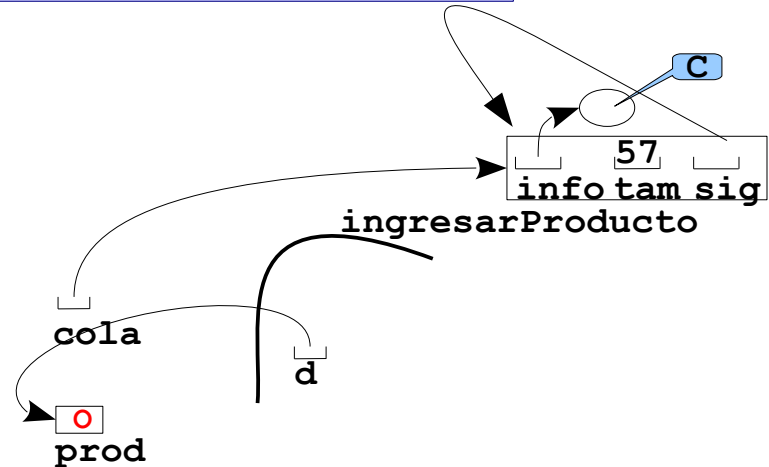
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

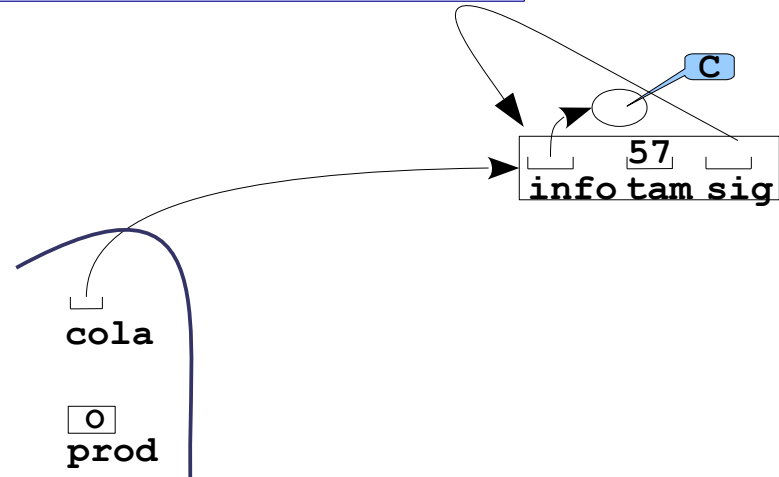


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

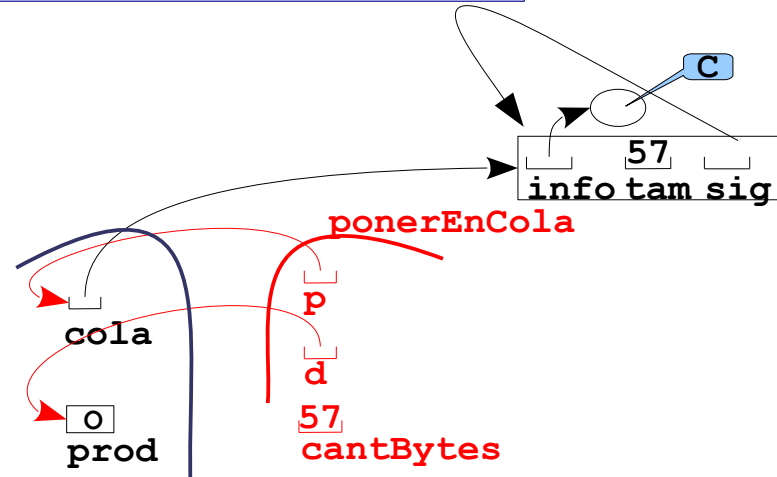


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

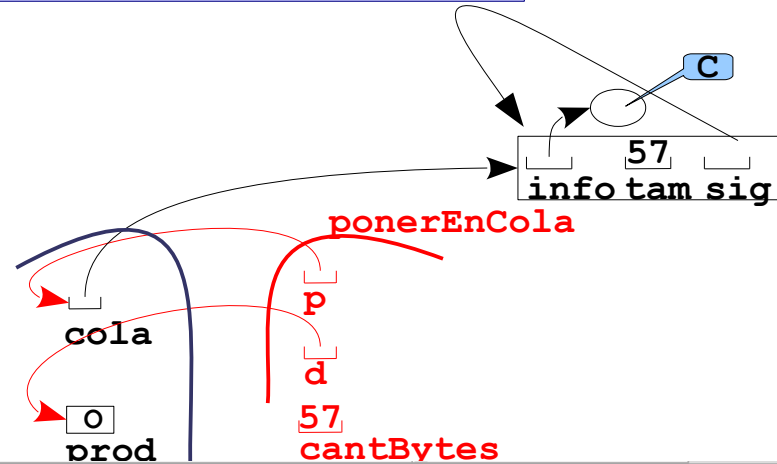


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

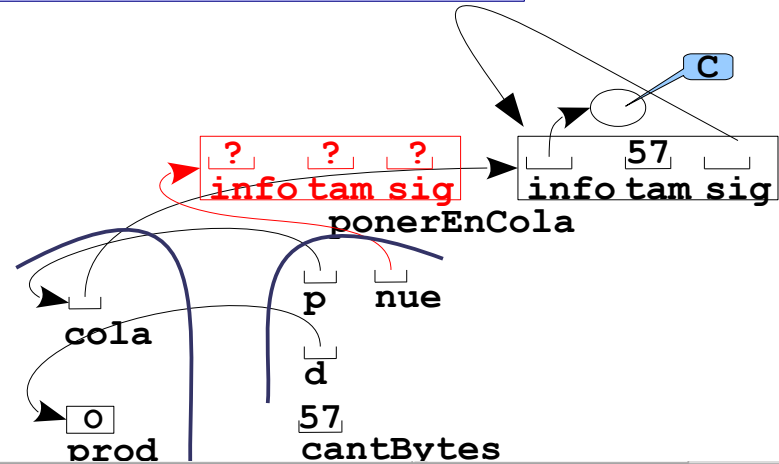


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

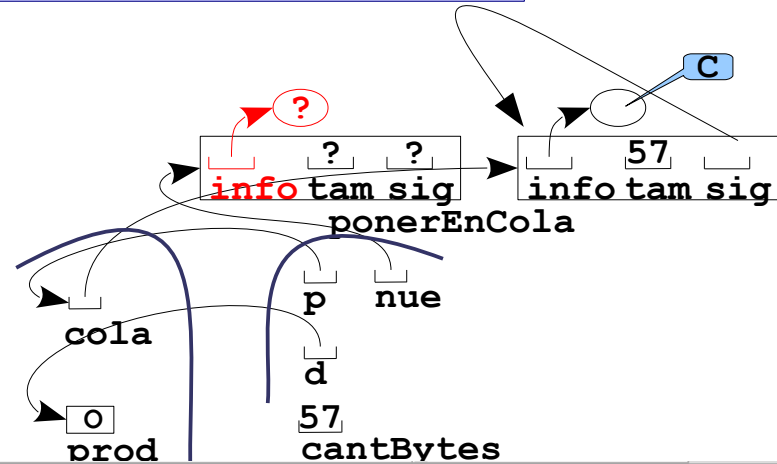


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
27        (nue->info = malloc(cantBytes)) == NULL)
28     {
29         free(nue);
30         return 0;
31     }
32     memcpy(nue->info, d, cantBytes);
33     nue->tamInfo = cantBytes;
34     if(*p == NULL)
35         nue->sig = nue;
36     else
37     {
38         nue->sig = (*p)->sig;
39         (*p)->sig = nue;
40     }
41     *p = nue;
42     return 1;
43 }
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

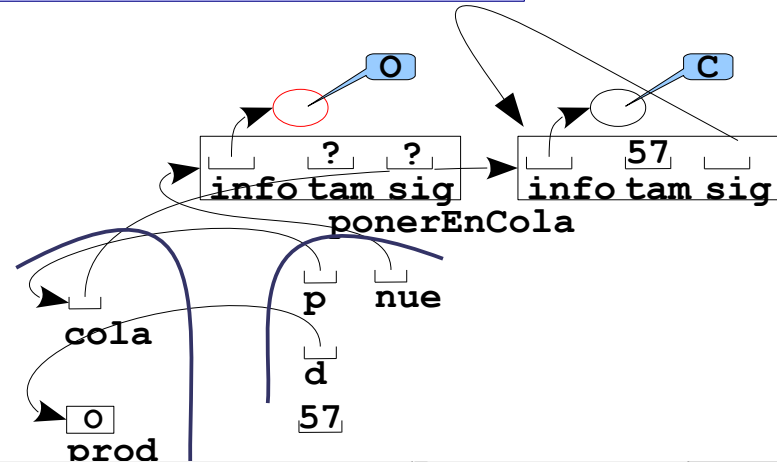


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

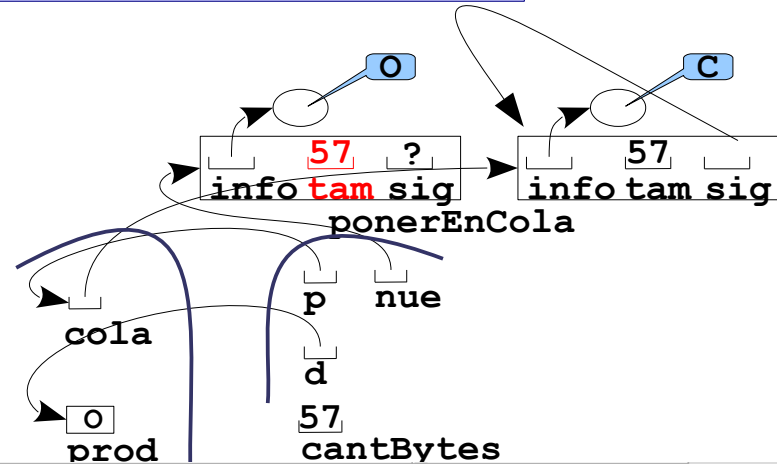


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



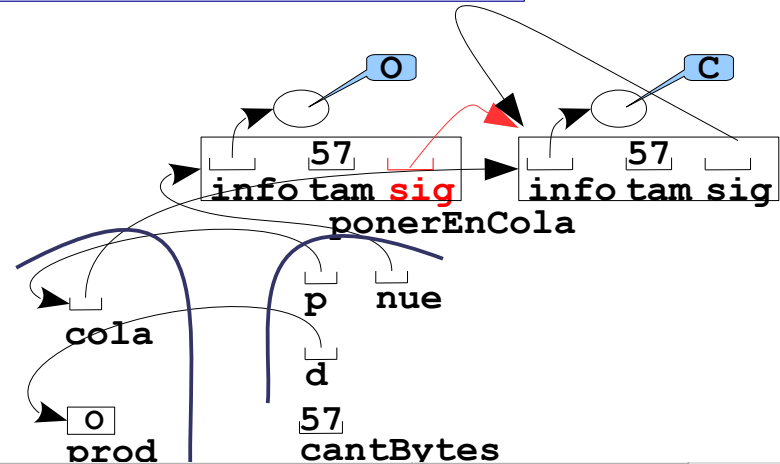
```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

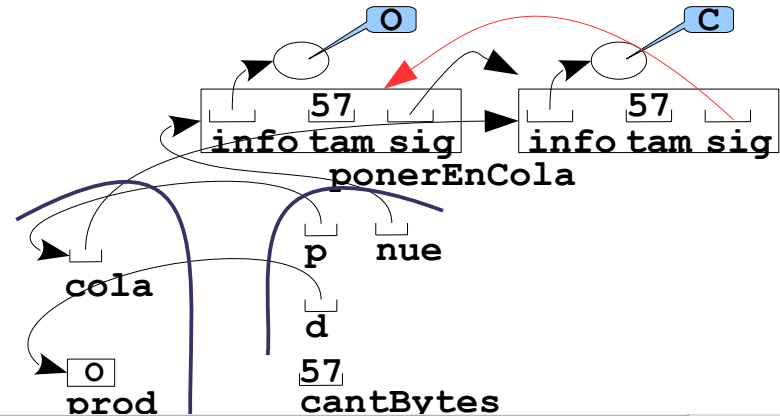


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

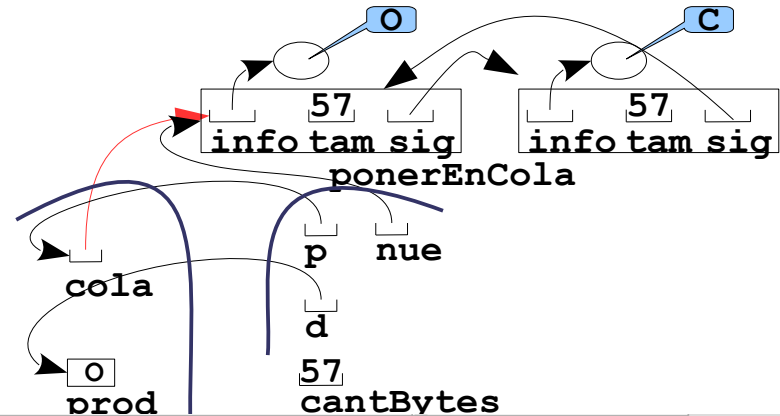


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

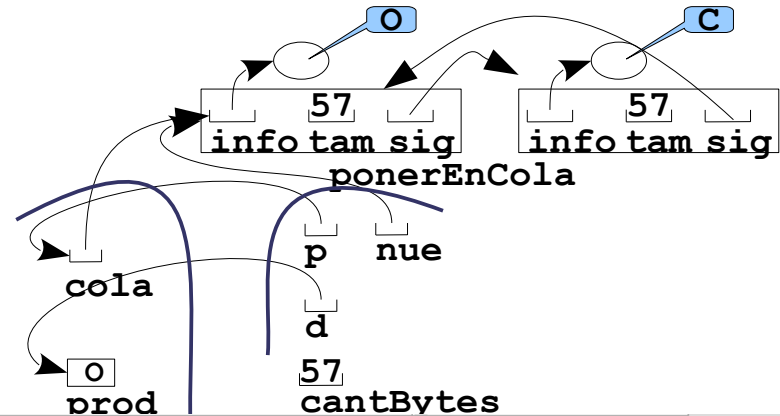


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

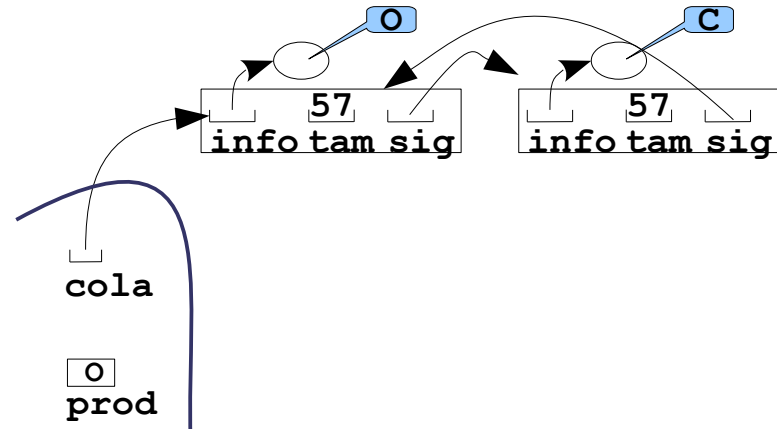


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

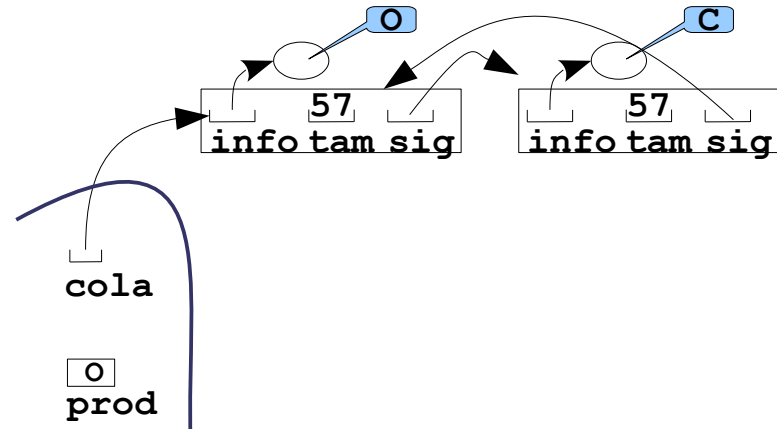


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

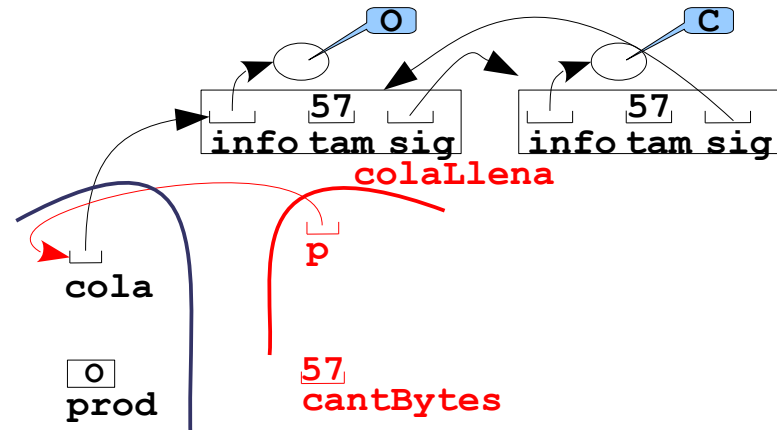


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

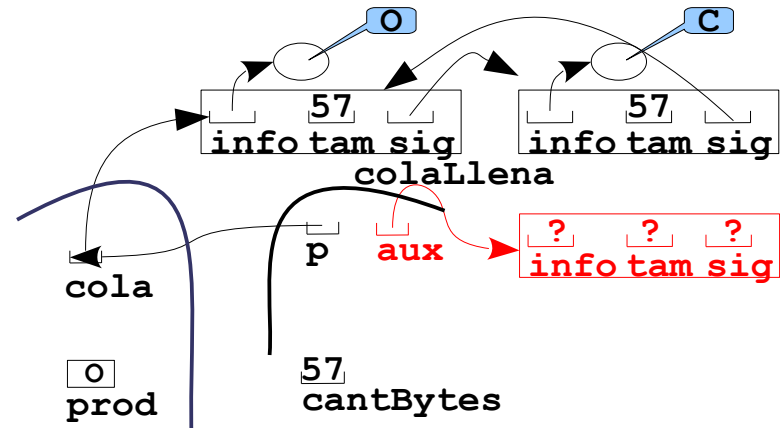


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



```
cola.c X cola.h X *main.c X main.h X
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c X cola.h X *main.c X main.h X
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```



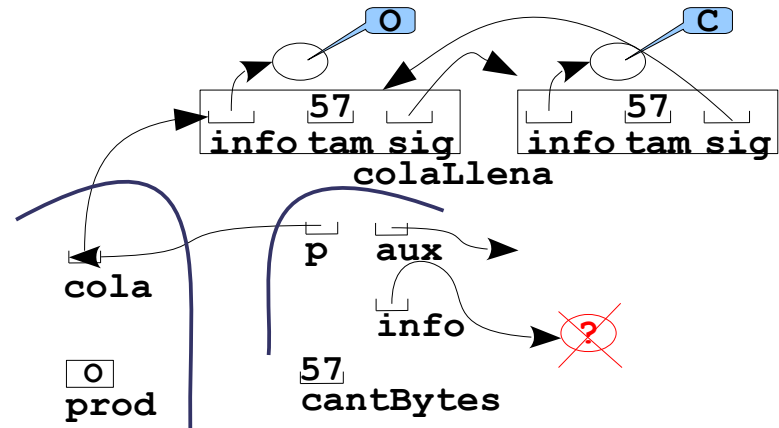




# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



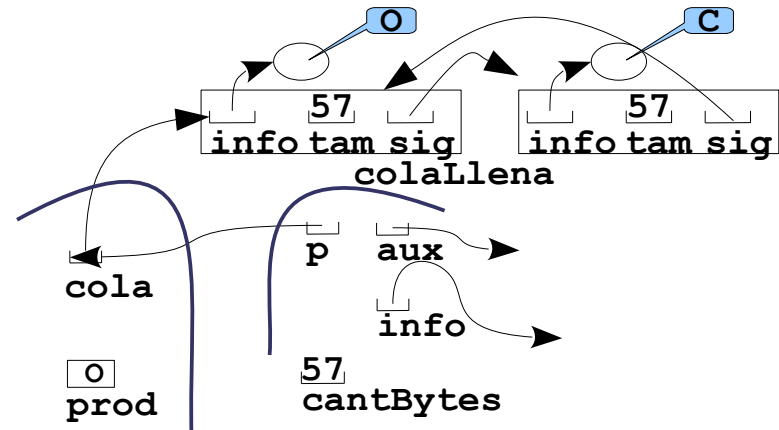
```
cola.c X cola.h X *main.c X main.h X
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c X cola.h X *main.c X main.h X
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



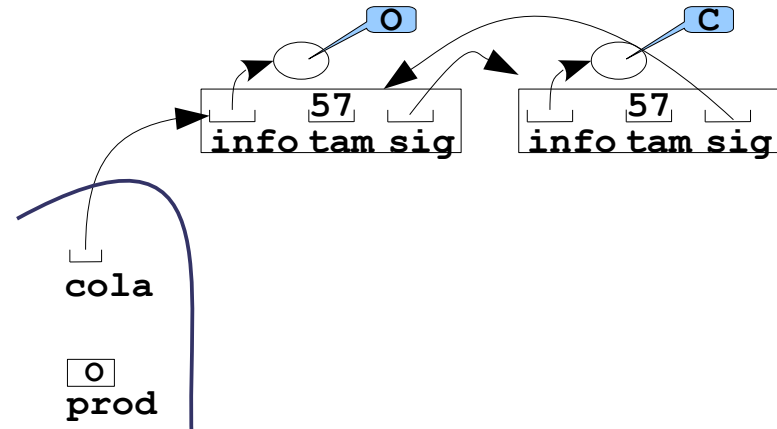
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;

cola.c x cola.h x *main.c x main.h x
13 int colaLlena(const tCola *p, unsigned cantBytes)
14 {
15     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
16     void *info = malloc(cantBytes);
17
18     free(aux);
19     free(info);
20     return aux == NULL || info == NULL;
21 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

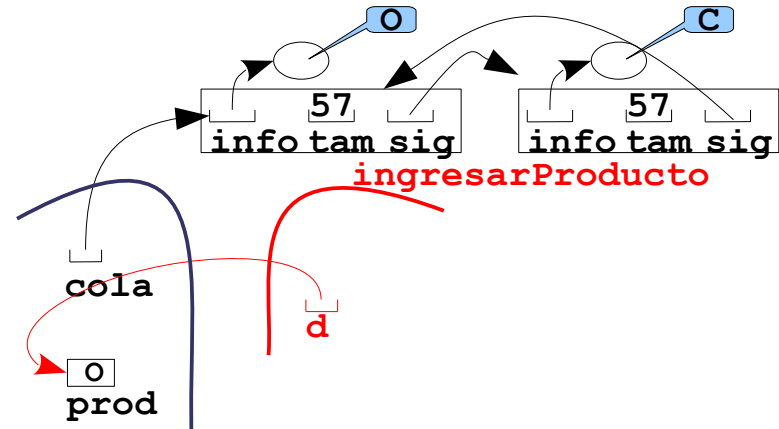


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while (!colaLlena(cola, sizeof(prod)) || ingresarProducto(&prod))
61     {
62         if (!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

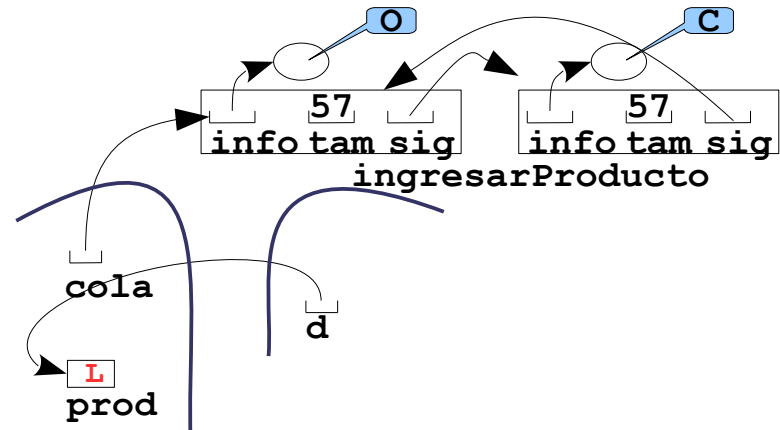


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

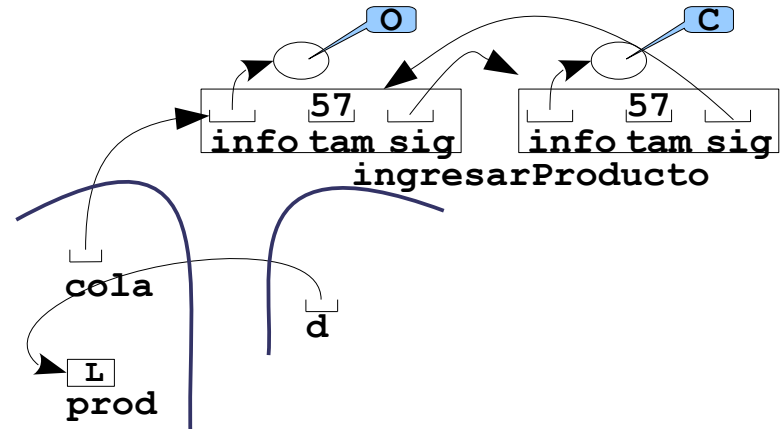


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



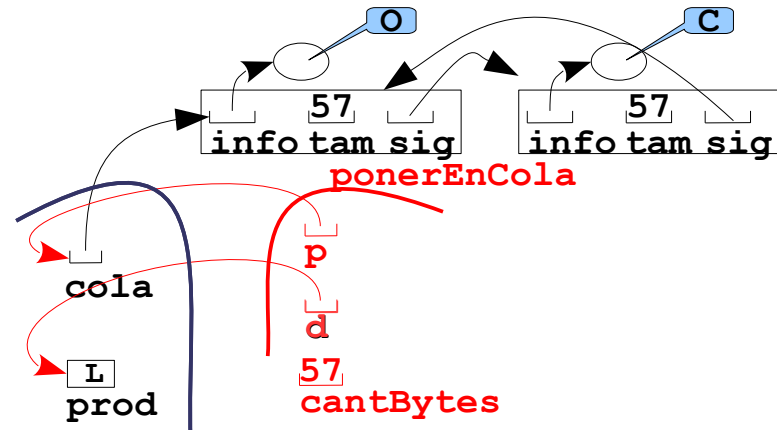
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```



# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

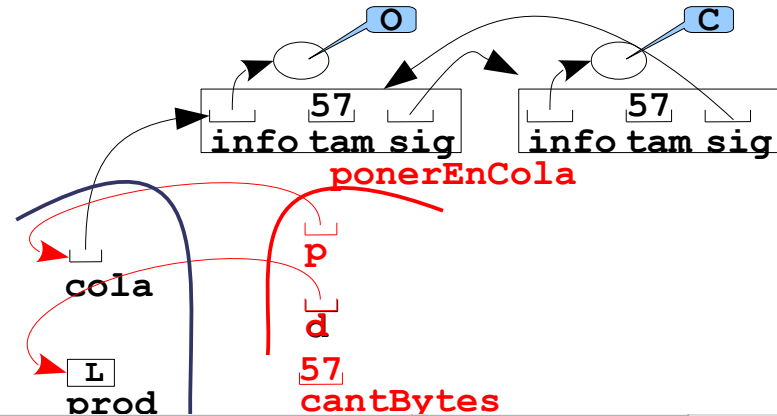


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

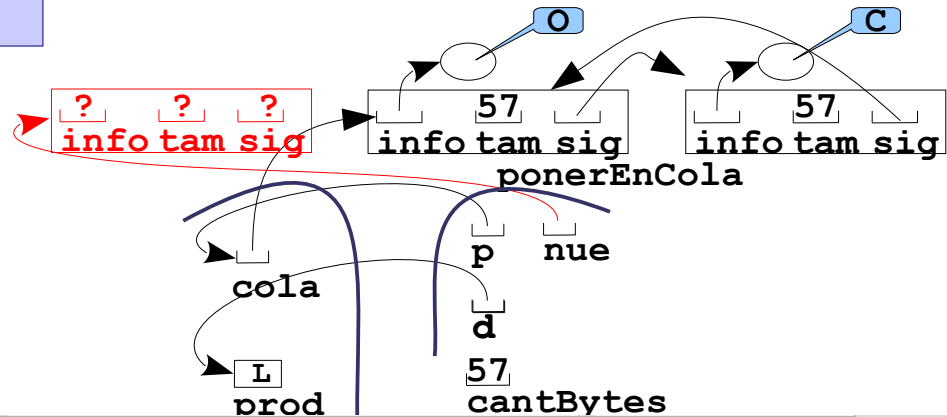


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

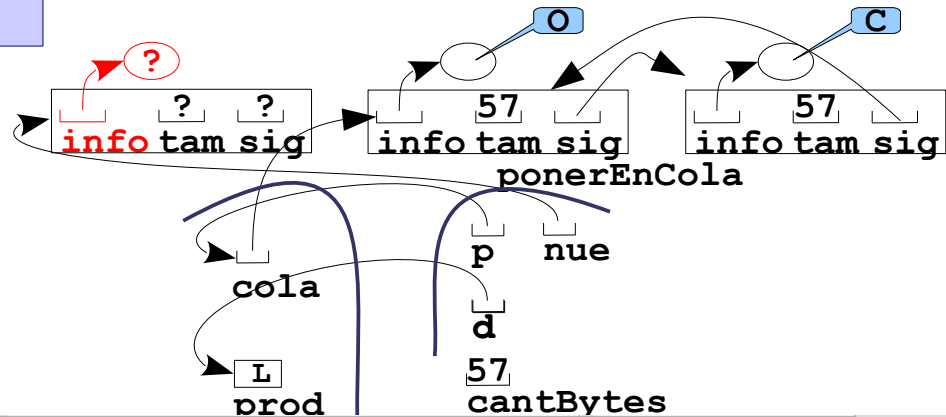


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
27        (nue->info = malloc(cantBytes)) == NULL)
28     {
29         free(nue);
30         return 0;
31     }
32     memcpy(nue->info, d, cantBytes);
33     nue->tamInfo = cantBytes;
34     if(*p == NULL)
35         nue->sig = nue;
36     else
37     {
38         nue->sig = (*p)->sig;
39         (*p)->sig = nue;
40     }
41     *p = nue;
42     return 1;
43 }
44 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



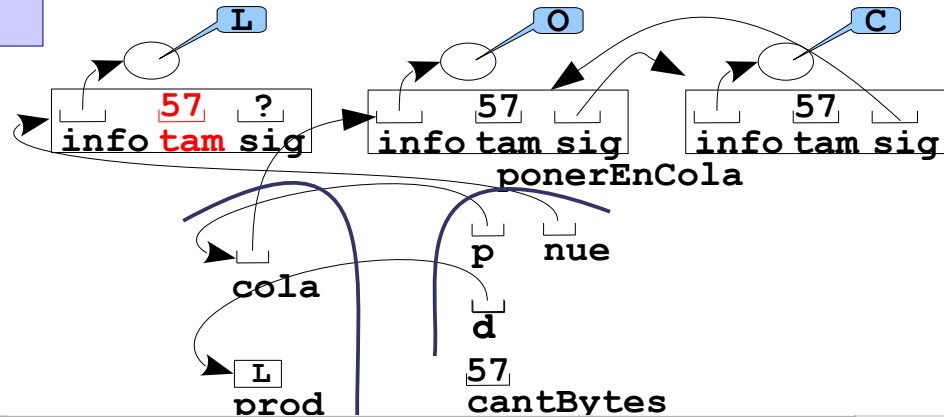
```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if ((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28         (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if (*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```



# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

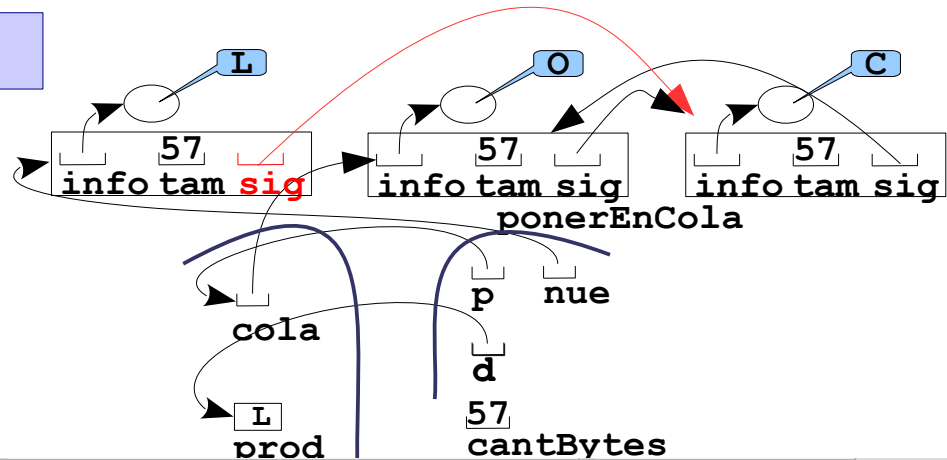


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

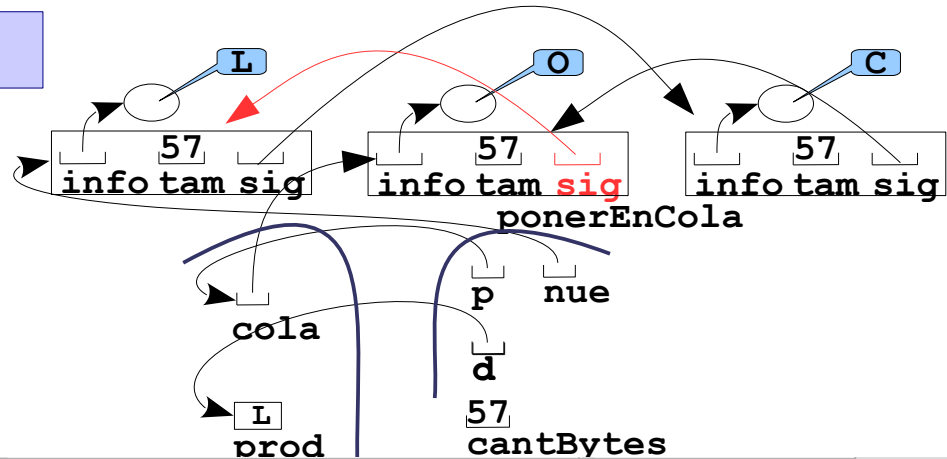


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



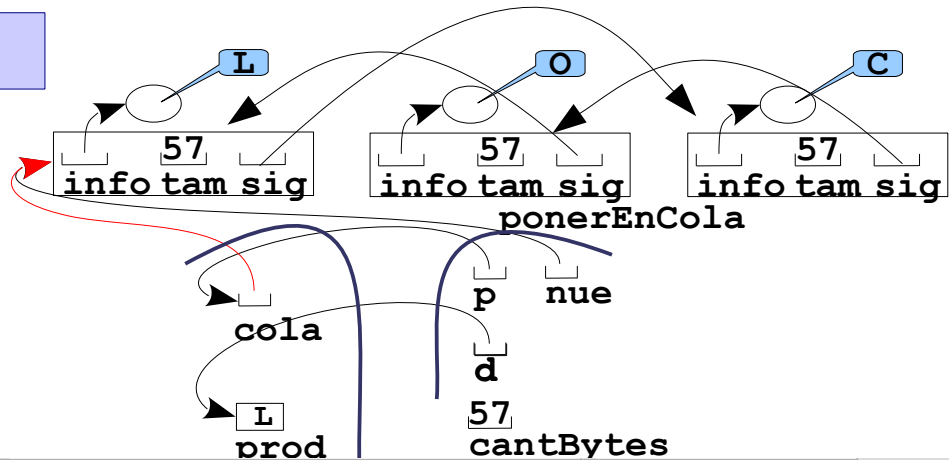
```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

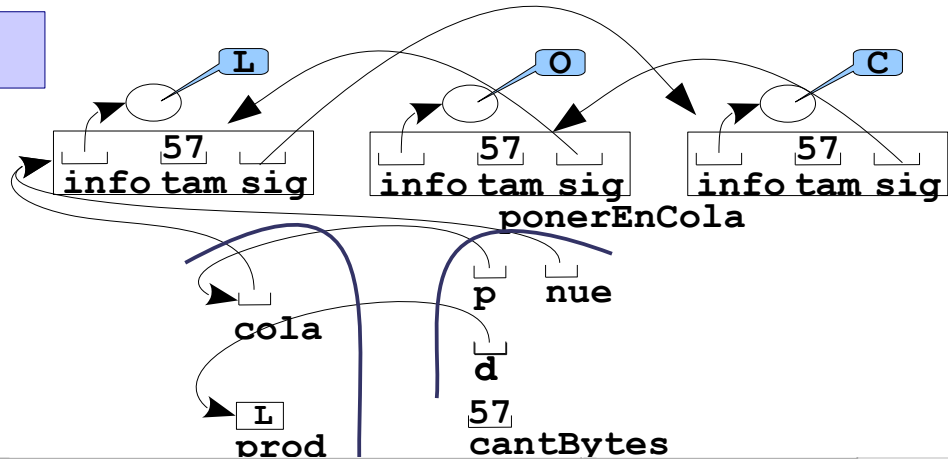


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

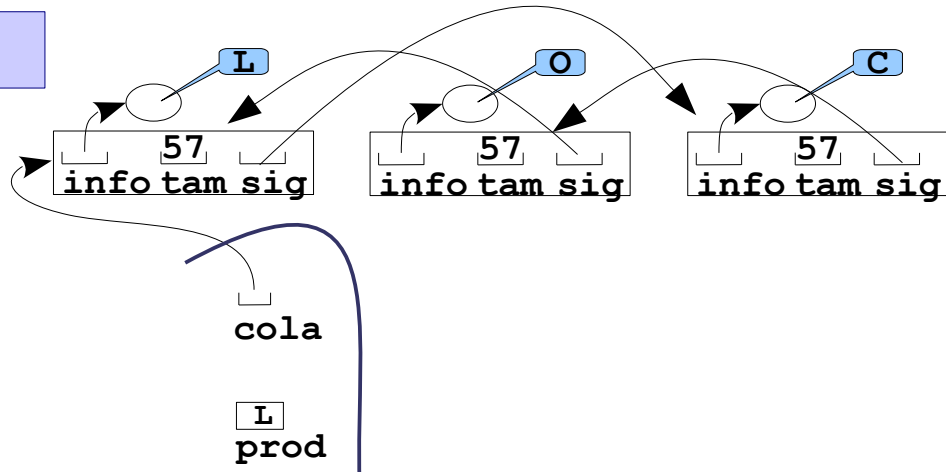


```
cola.c x cola.h x *main.c x main.h x
23 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
24 {
25     tNodo *nue;
26
27     if((nue = (tNodo *)malloc(sizeof(tNodo))) == NULL ||
28        (nue->info = malloc(cantBytes)) == NULL)
29     {
30         free(nue);
31         return 0;
32     }
33     memcpy(nue->info, d, cantBytes);
34     nue->tamInfo = cantBytes;
35     if(*p == NULL)
36         nue->sig = nue;
37     else
38     {
39         nue->sig = (*p)->sig;
40         (*p)->sig = nue;
41     }
42     *p = nue;
43     return 1;
44 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

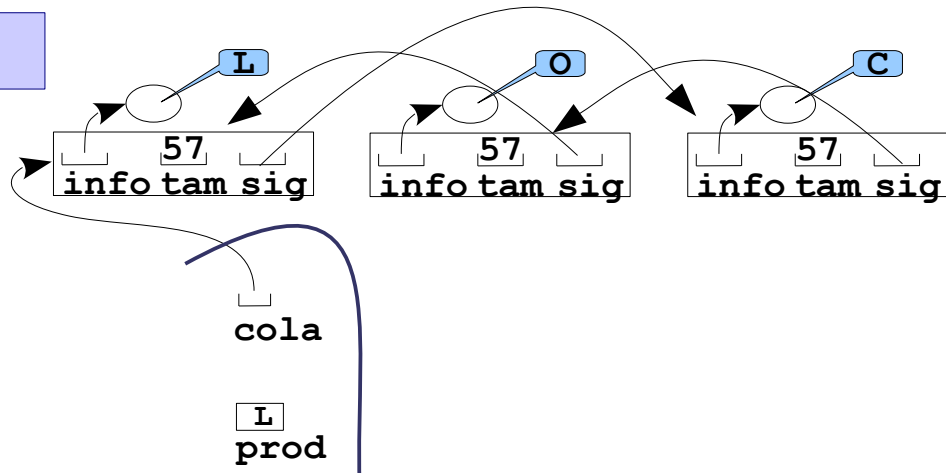


```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67             mostrarProducto(&prod);
68             cant++;
69         }
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



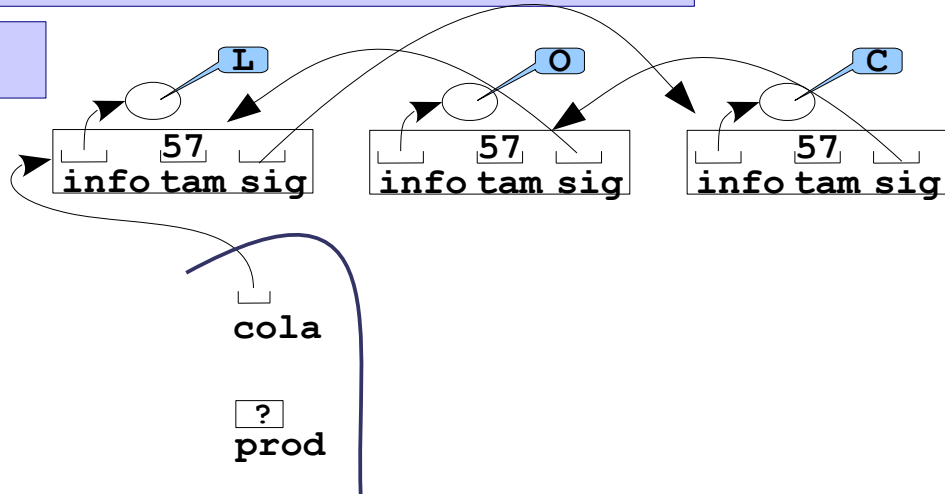
```
cola.c x cola.h x *main.c x main.h x
53 int _probarLlenaYEncolar(tCola *cola)
54 {
55     tProd | prod;
56     int    cant = 0;
57
58     puts("Probando cola llena y poner en cola.");
59     mostrarProducto(NULL);
60     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
61     {
62         if(!ponerEnCola(cola, &prod, sizeof(prod)))
63         {
64             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
65             puts("no se pudo cargar la informacion y"
66                 " habria que tomar alguna decision drastica.");
67         }
68         mostrarProducto(&prod);
69         cant++;
70     }
71     return cant;
72 }
```

# Tipos de Datos Abstractos

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

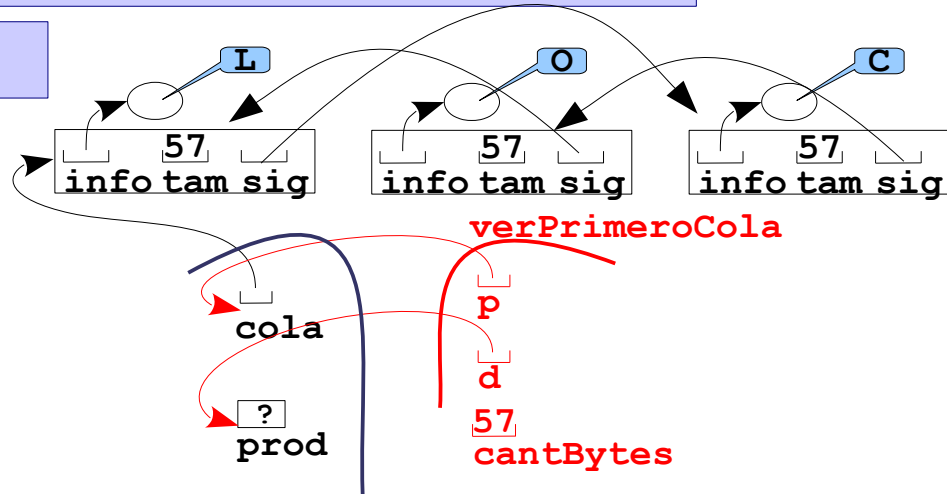


```
cola.c x cola.h x *main.c x main.h x
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))
81     {
82         mostrarProducto(NULL);
83         mostrarProducto(&prod);
84     }
85     else
86         puts("La cola estaba vacia.");
87     puts("");
88 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

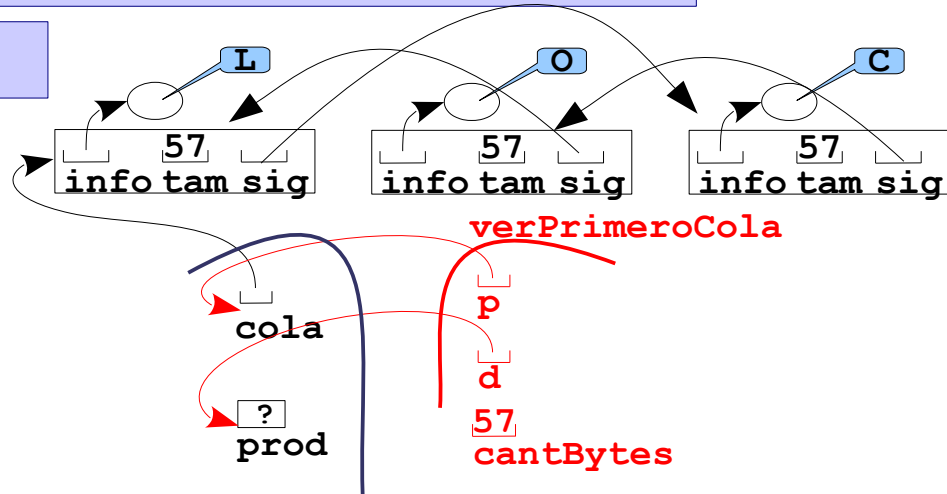


```
cola.c x cola.h x *main.c x main.h x
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))
81     {
82         mostrarProducto(NULL);
83         mostrarProducto(&prod);
84     }
85     else
86         puts("La cola estaba vacia.");
87     puts("");
88 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c X cola.h X *main.c X main.h X
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))

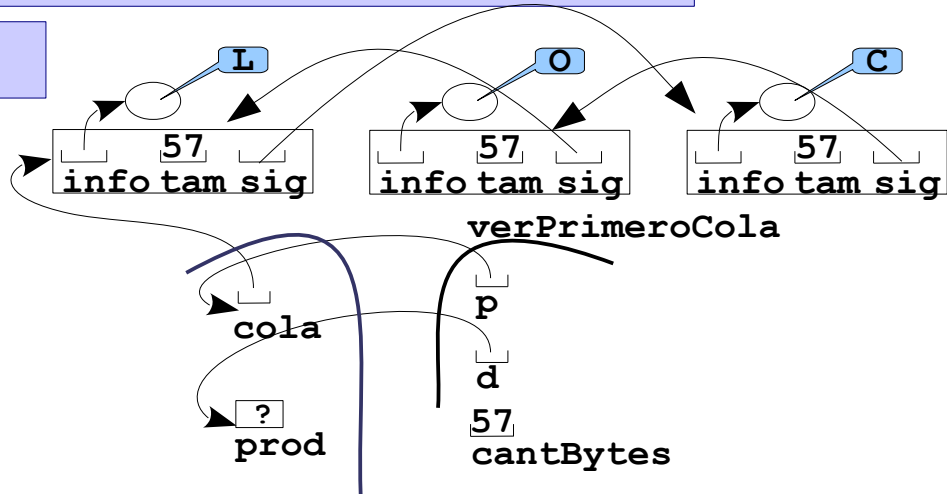
cola.c X cola.h X *main.c X main.h X
46 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes)
47 {
48     if(*p == NULL)
49         return 0;
50     memcpy(d, (*p)->sig->info, minimo(cantBytes, (*p)->sig->tamInfo));
51     return 1;
52 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



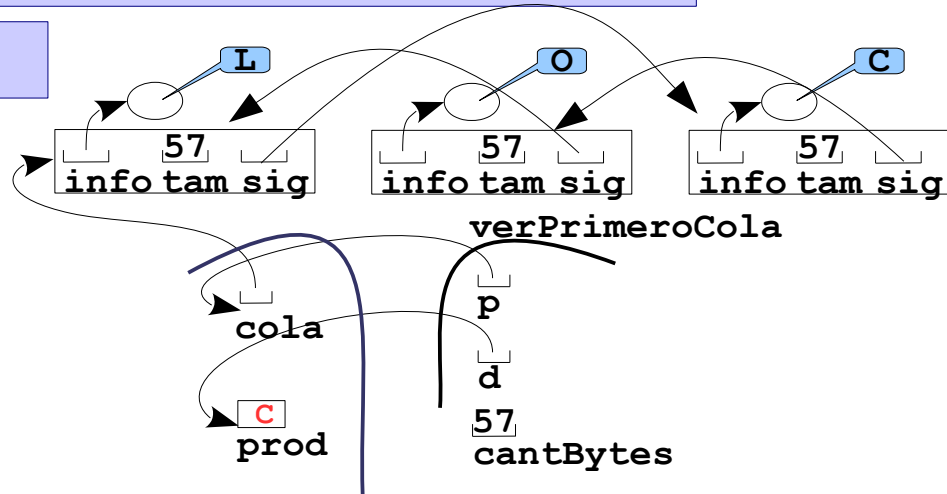
```
cola.c X cola.h X *main.c X main.h X
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))
```

```
cola.c X cola.h X *main.c X main.h X
46 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes)
47 {
48     if(*p == NULL)
49         return 0;
50     memcpy(d, (*p)->sig->info, minimo(cantBytes, (*p)->sig->tamInfo));
51     return 1;
52 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



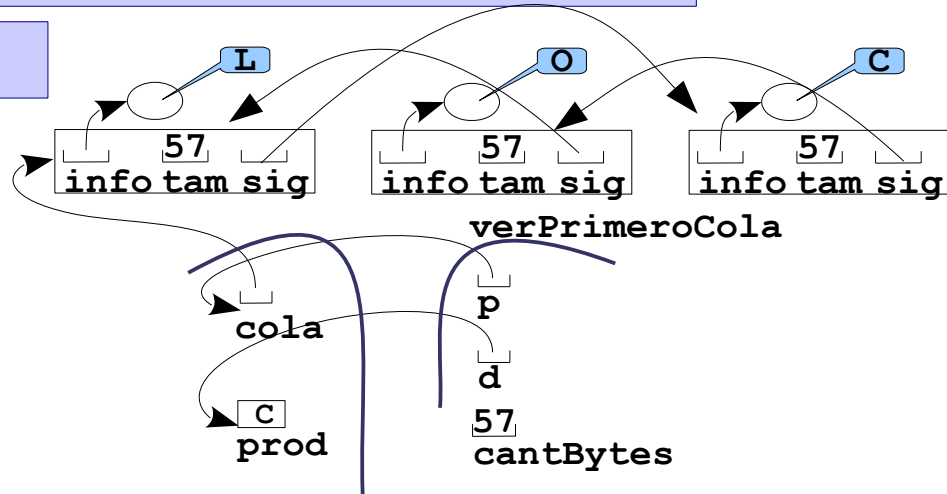
```
cola.c x cola.h x *main.c x main.h x
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))
```

```
cola.c x cola.h x *main.c x main.h x
46 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes)
47 {
48     if(*p == NULL)
49         return 0;
50     memcpy(d, (*p)->sig->info, minimo(cantBytes, (*p)->sig->tamInfo));
51     return 1;
52 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



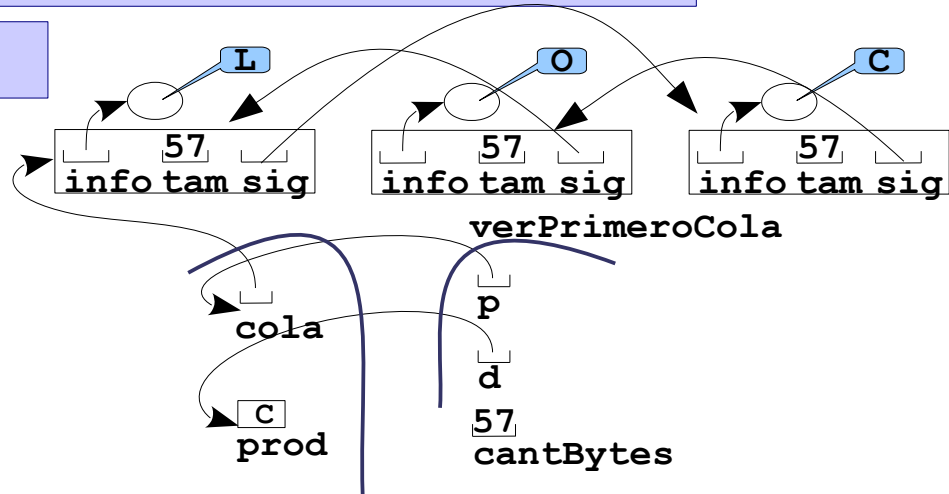
```
cola.c X cola.h X *main.c X main.h X
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))
```

```
cola.c X cola.h X *main.c X main.h X
46 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes)
47 {
48     if(*p == NULL)
49         return 0;
50     memcpy(d, (*p)->sig->info, minimo(cantBytes, (*p)->sig->tamInfo));
51     return 1;
52 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



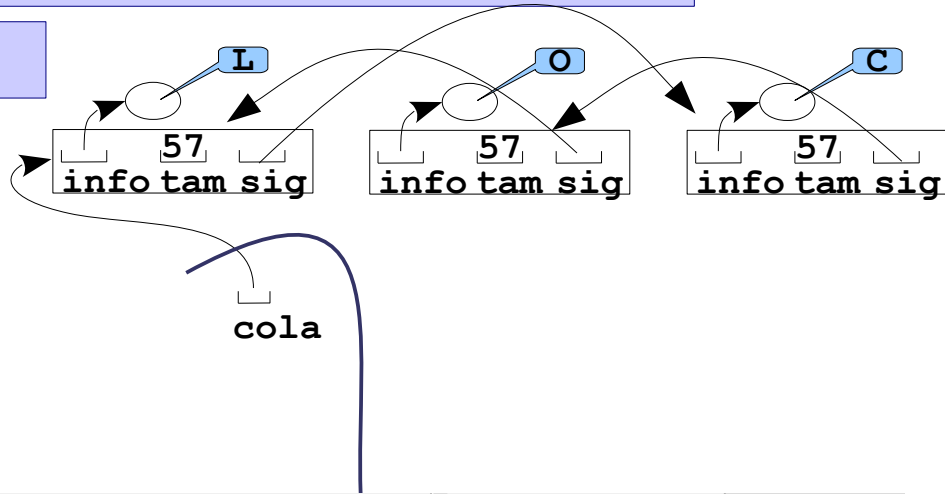
```
cola.c x cola.h x *main.c x main.h x
75 void _probarVerTope(tCola *cola)
76 {
77     tProd prod;
78
79     puts("Probando ver el primero de la cola.");
80     if(verPrimeroCola(col, &prod, sizeof(prod)))
81     {
82         mostrarProducto(NULL);
83         mostrarProducto(&prod);
84     }
85     else
86         puts("La cola estaba vacia.");
87     puts("");
88 }
```



# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

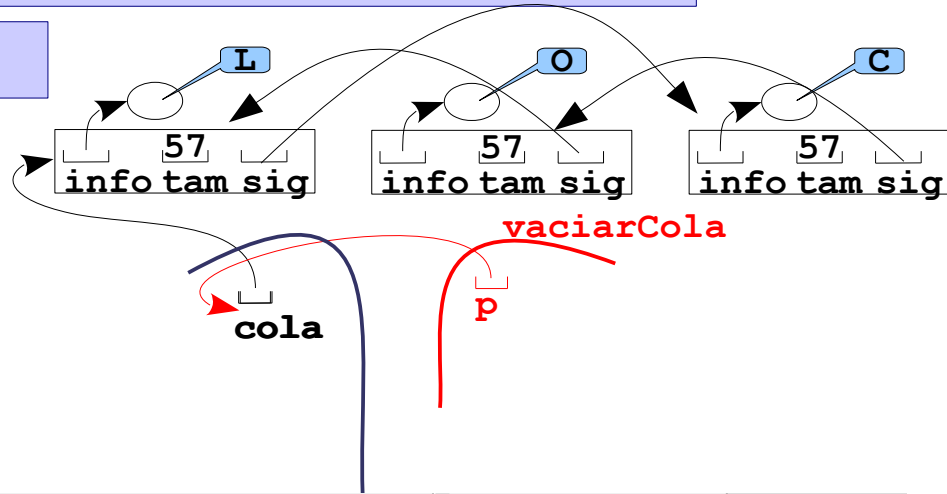


```
cola.c X cola.h X *main.c X main.h X
110 int _probarVaciarColaYColaVacía(tCola *cola)
111 {
112     puts("Probando vaciar cola y cola vacía.");
113     vaciarCola (cola);
114     if (!colaVacía (cola))
115         return 0; // fprintf(stderr, "ERROR - la cola debía estar vacía\n");
116     printf("Vaciar cola funciona!\n\n");
117     puts("");
118     return 1;
119 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

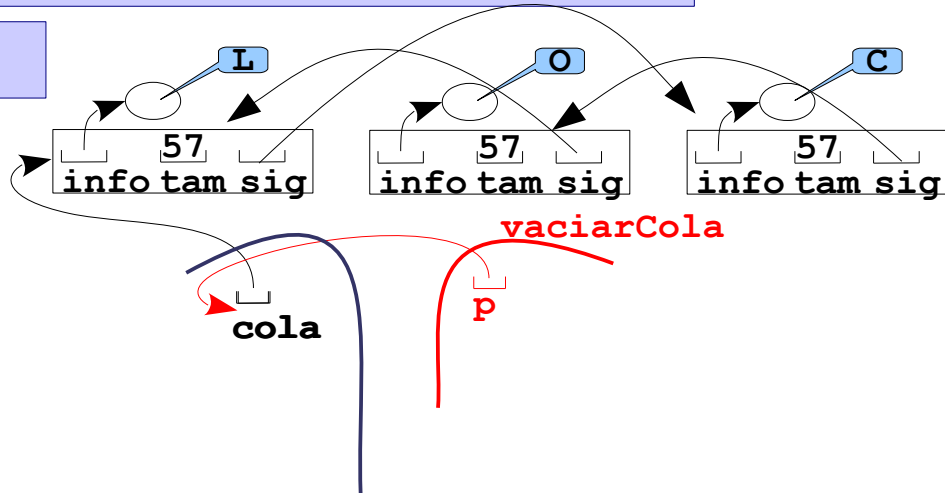


```
cola.c X cola.h X *main.c X main.h X
110 int _probarVaciarColaYColaVacia(tCola *cola)
111 {
112     puts("Probando vaciar cola y cola vacia.");
113     vaciarCola(col);
114     if(!colaVacia(col))
115         return 0; // fprintf(stderr, "ERROR - la cola debia estar vacia\n");
116     printf("Vaciar cola funciona!\n\n");
117     puts("");
118     return 1;
119 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

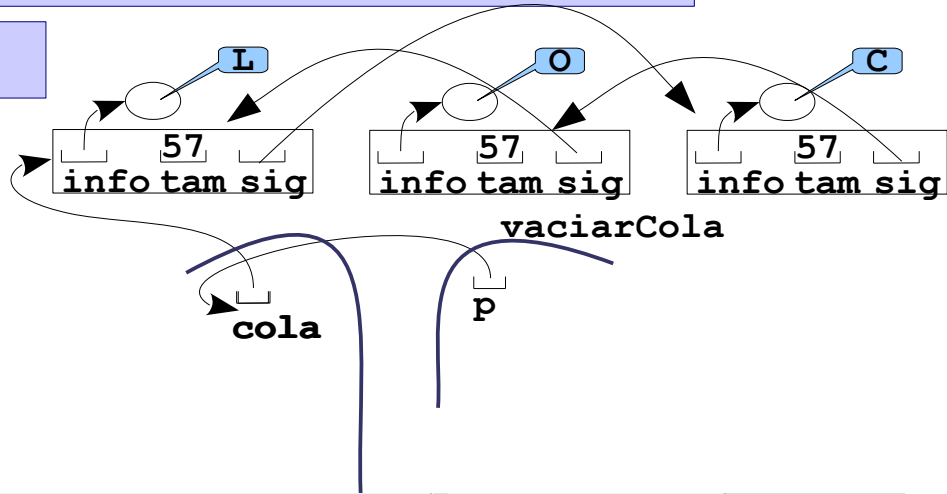
cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



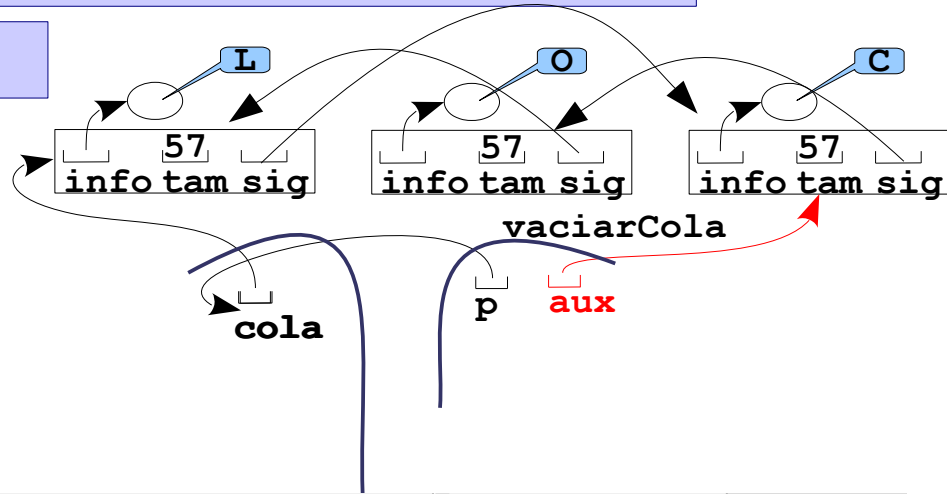
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodeo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

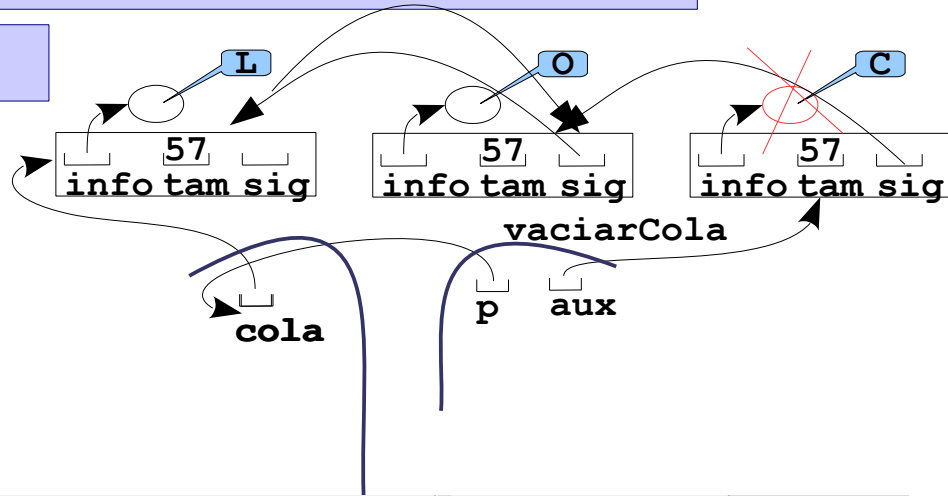
cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



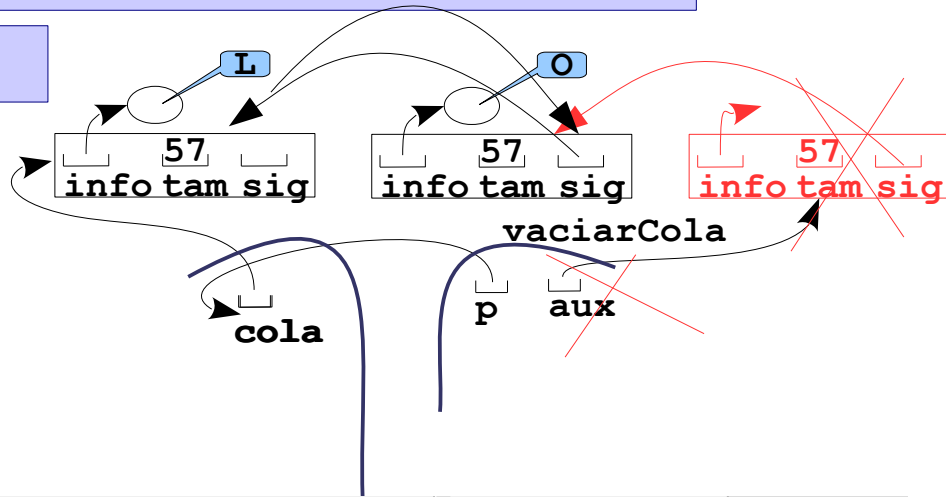
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



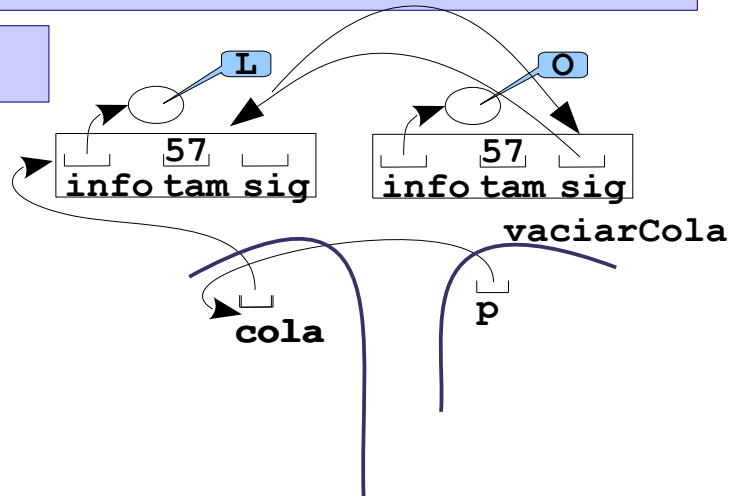
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }
```

```
cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNode *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



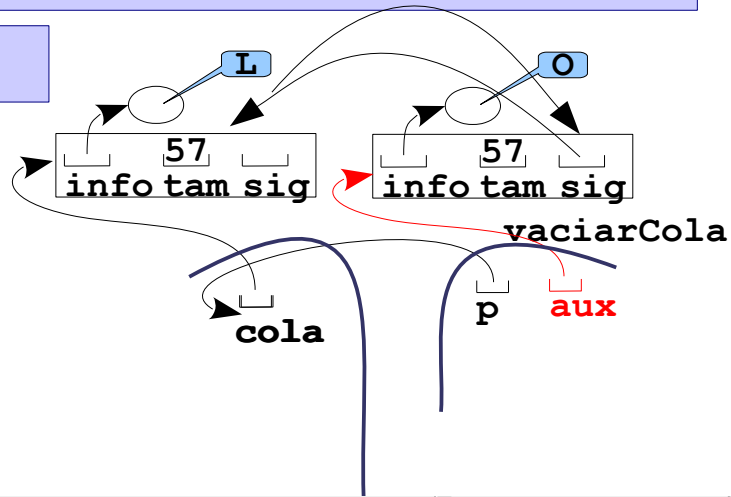
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }
```

```
cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNode *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



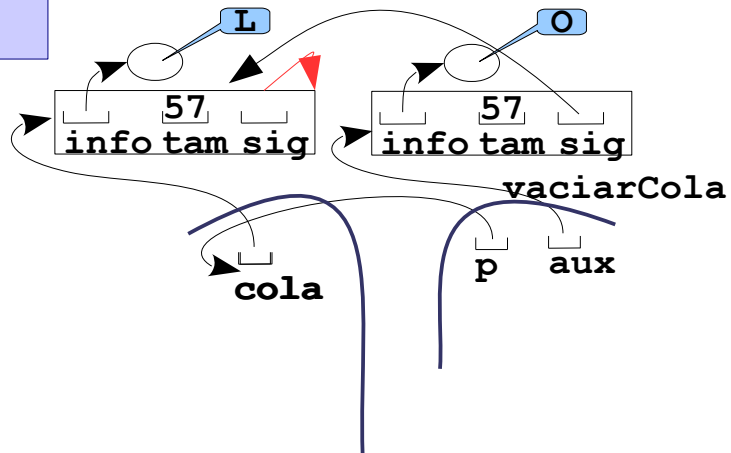
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }
```

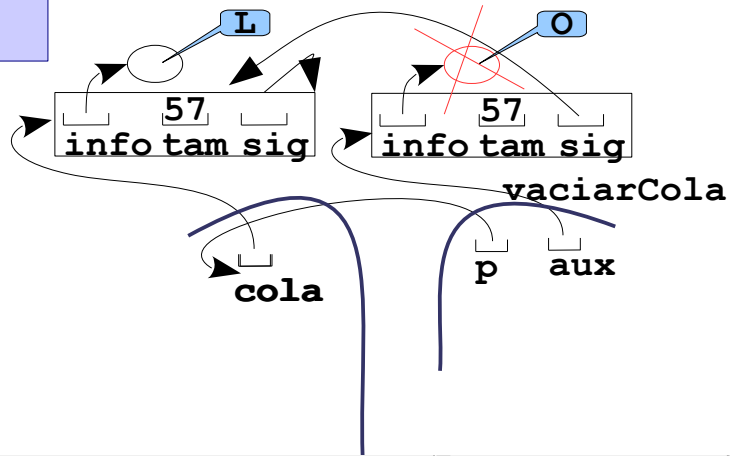
```
cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNode *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```



# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



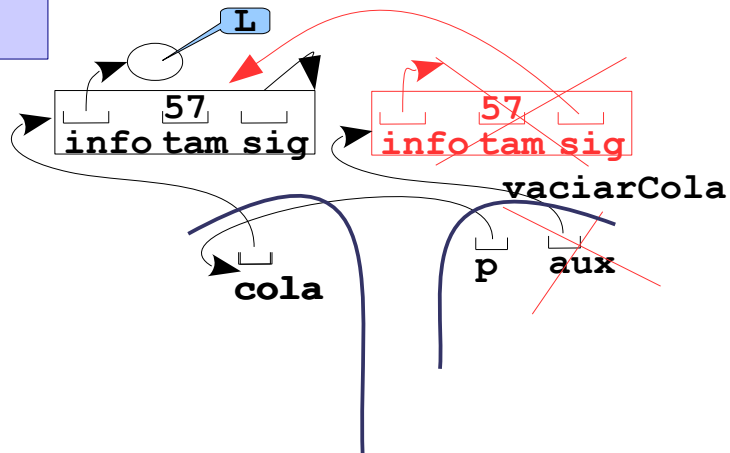
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNode *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



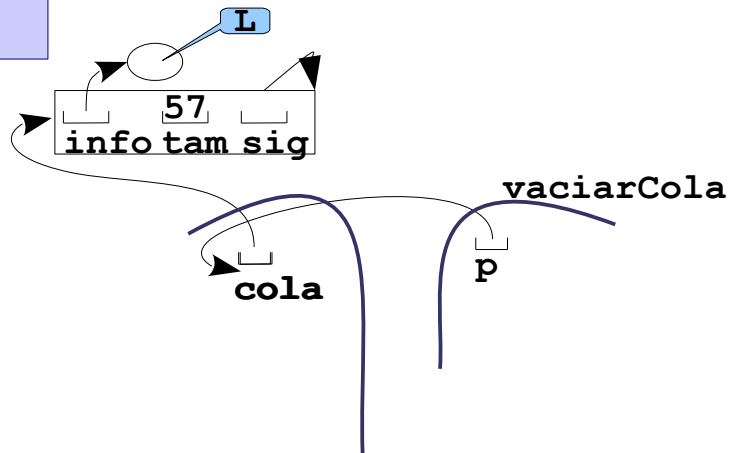
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNode *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



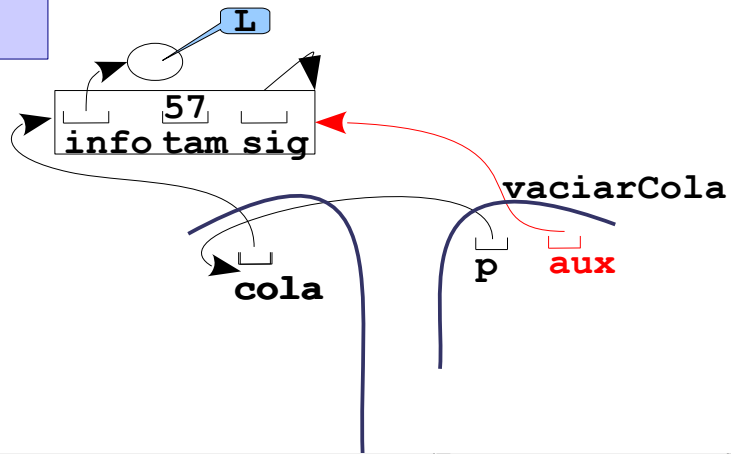
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



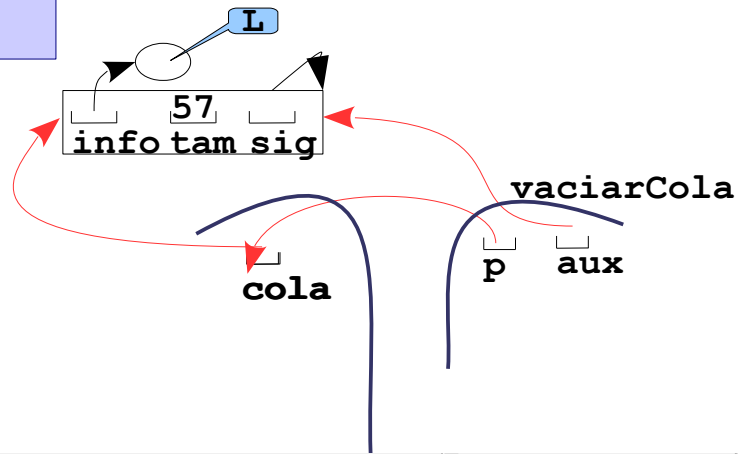
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



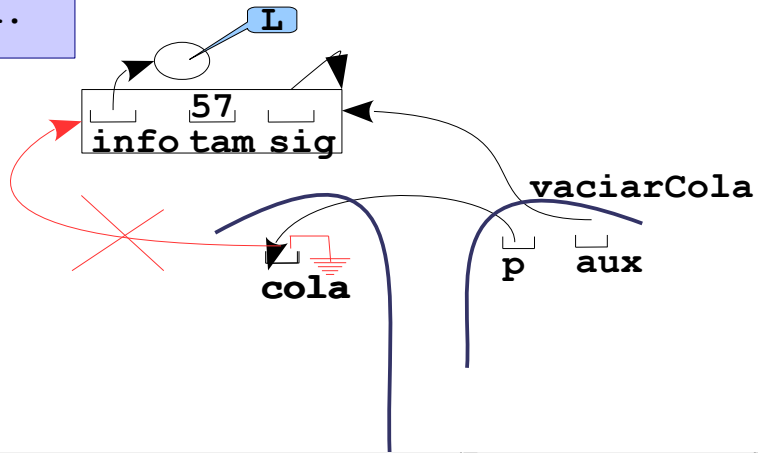
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



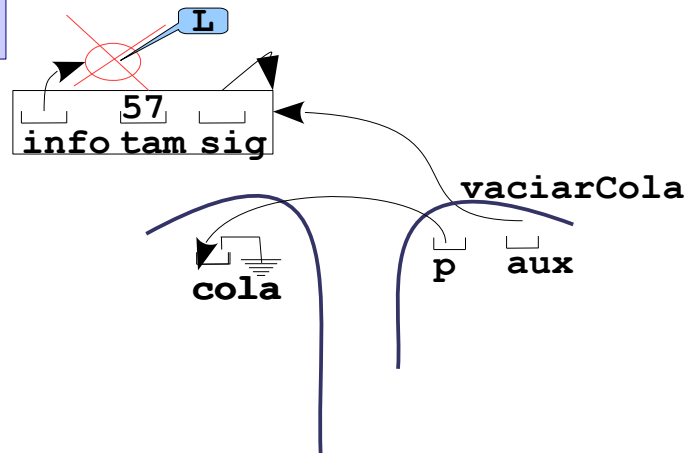
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaciarCola()
111 {
112     puts("Probar vaciar Cola.");
113     vaciarCola(&cola);
114     if(!colaVacio())
115         return 0;
116     printf("Vaciar Cola exitoso.\n");
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



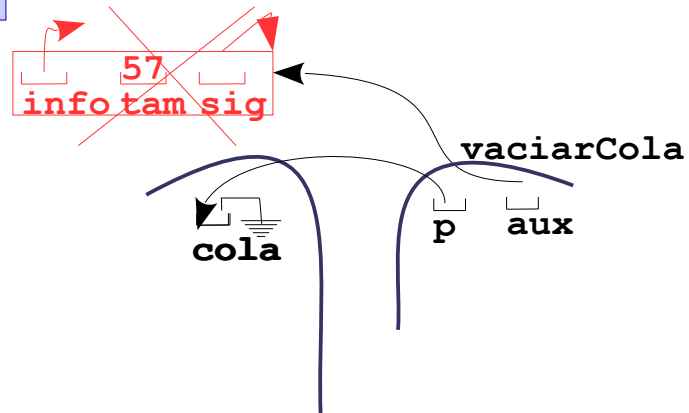
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaciarCola()
111 {
112     puts("Probar vaciar cola");
113     vaciarCola(&cola);
114     if(!colaVacio())
115         return 0;
116     printf("Vaciar cola exitosa\n");
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaci
111 {
112     puts("Proba
113     vaciarCola(
114     if(!colaVac
115         return
116     printf("Vac
117     puts("");
118     return 1;
119 }

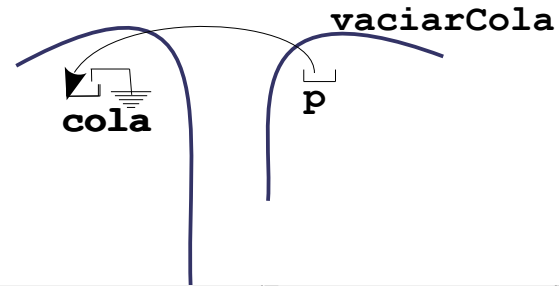
cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaciarCola()
111 {
112     puts("Probar vaciarCola");
113     vaciarCola(&cola);
114     if(!colaVacio())
115         return 0;
116     printf("VaciarCola exitoso\n");
117     puts("");
118     return 1;
119 }

cola.c x cola.h x *main.c x main.h x
76 void vaciarCola(tCola *p)
77 {
78     while(*p)
79     {
80         tNodo *aux = (*p)->sig;
81
82         if(*p == aux)
83             *p = NULL;
84         else
85             (*p)->sig = aux->sig;
86         free(aux->info);
87         free(aux);
88     }
89 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

  
cola

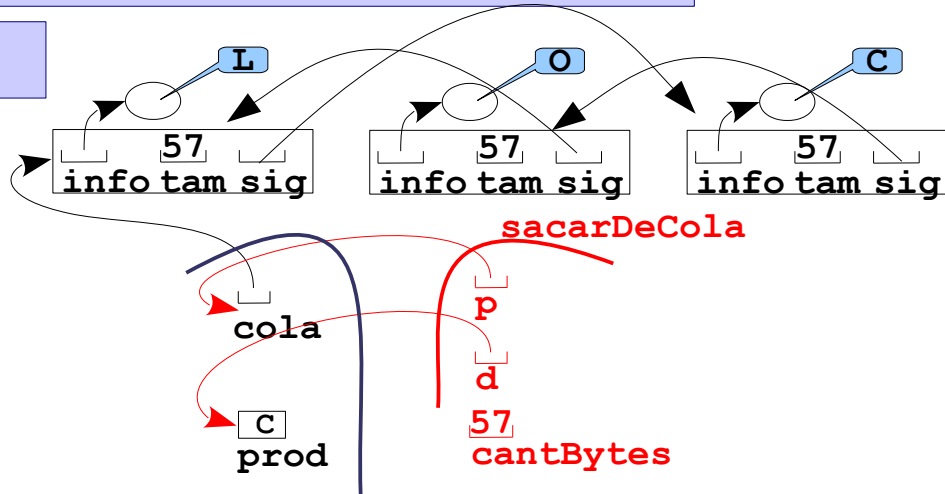
```
cola.c x cola.h x *main.c x main.h x
110 int _probarVaciarColaYColaVacía (tCola *cola)
111 {
112     puts("Probando vaciar cola y cola vacía.");
113     _vaciarCola (cola);
114     if (!_colaVacía (cola))
115         return 0; // fprintf(stderr, "ERROR - la cola debía estar vacía\n");
116     printf("Vaciar cola funciona!\n\n");
117     puts("");
118     return 1;
119 }
```

# Tipos de Datos Abstractos

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

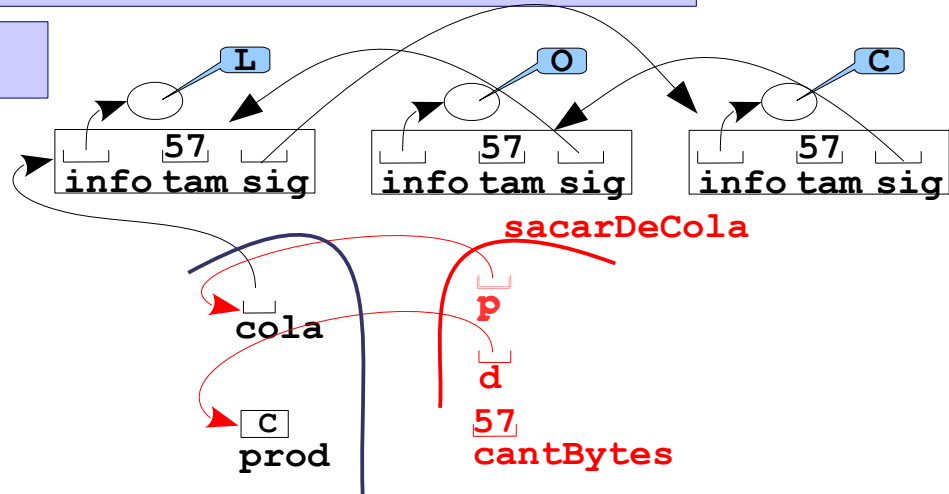


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(col, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

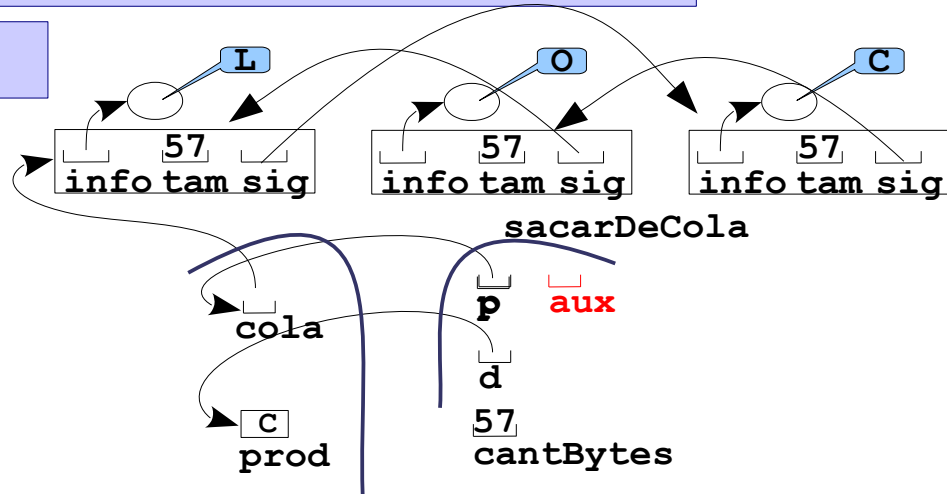


```
cola.c X cola.c X cola.h X *main.c X main.h X
101 59 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102 60 {
103 61     tNodo *aux;
104 62
105 63     if(*p == NULL)
106 64         return 0;
107 65     aux = (*p)->sig;
108 66     memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109 67     if(aux == *p)
110 68         *p = NULL;
111 69     else
112 70         (*p)->sig = aux->sig;
113 71     free(aux->info);
114 72     free(aux);
115 73     return 1;
116 74 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

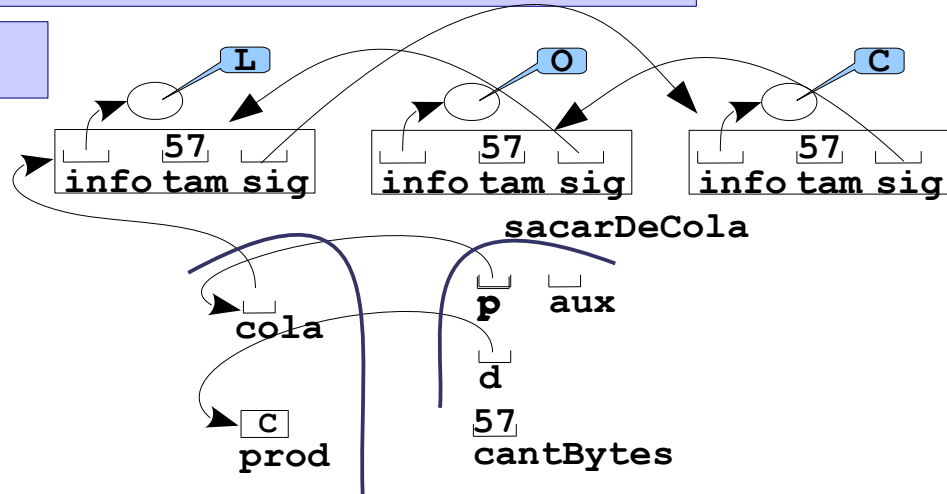


```
cola.c X cola.c X cola.h X *main.c X main.h X
101 59 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102 60 {
103 61     tNodo *aux;
104 62
105 63     if(*p == NULL)
106 64         return 0;
107 65     aux = (*p)->sig;
108 66     memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109 67     if(aux == *p)
110 68         *p = NULL;
111 69     else
112 70         (*p)->sig = aux->sig;
113 71     free(aux->info);
114 72     free(aux);
115 73     return 1;
116 74 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

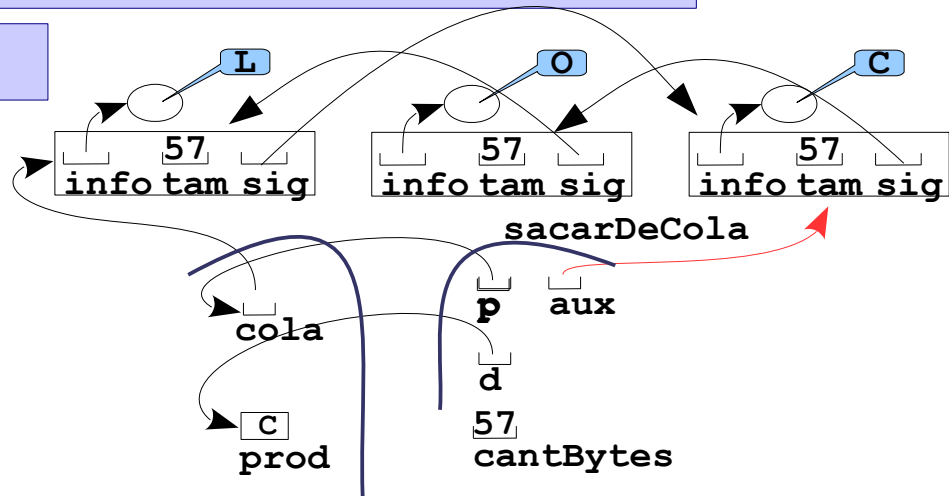


```
cola.c x cola.c x cola.h x *main.c x main.h x
101 59 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102 60 {
103 61     tNodo *aux;
104 62
105 63     if(*p == NULL)
106 64         return 0;
107 65     aux = (*p)->sig;
108 66     memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109 67     if(aux == *p)
110 68         *p = NULL;
111 69     else
112 70         (*p)->sig = aux->sig;
113 71     free(aux->info);
114 72     free(aux);
115 73     return 1;
116 74 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



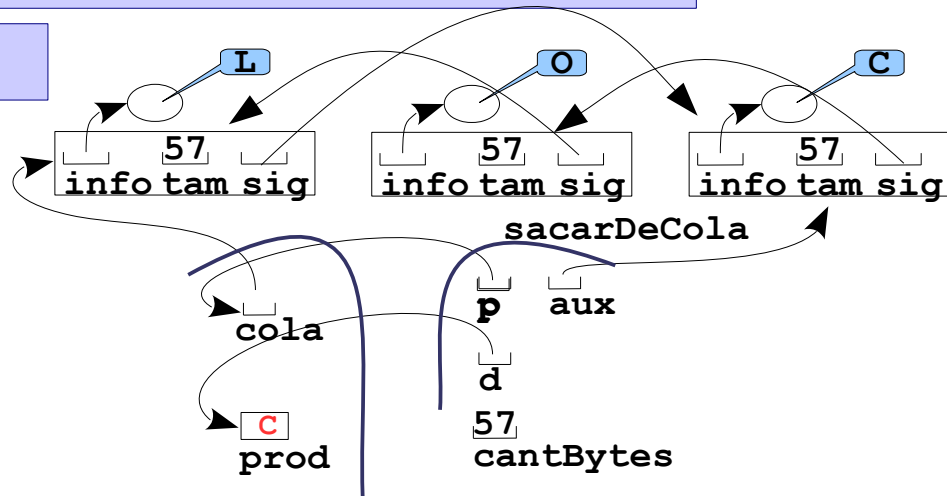
```
cola.c x cola.c x cola.h x *main.c x main.h x
101 59 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102 60 {
103 61     tNodo *aux;
104 62
105 63     if(*p == NULL)
106 64         return 0;
107 65     aux = (*p)->sig;
108 66     memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109 67     if(aux == *p)
110 68         *p = NULL;
111 69     else
112 70         (*p)->sig = aux->sig;
113 71     free(aux->info);
114 72     free(aux);
115 73     return 1;
116 74 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

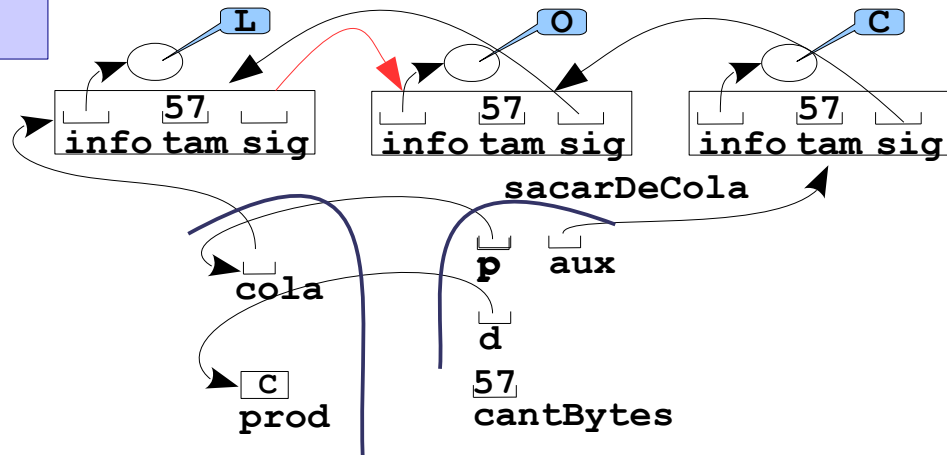


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

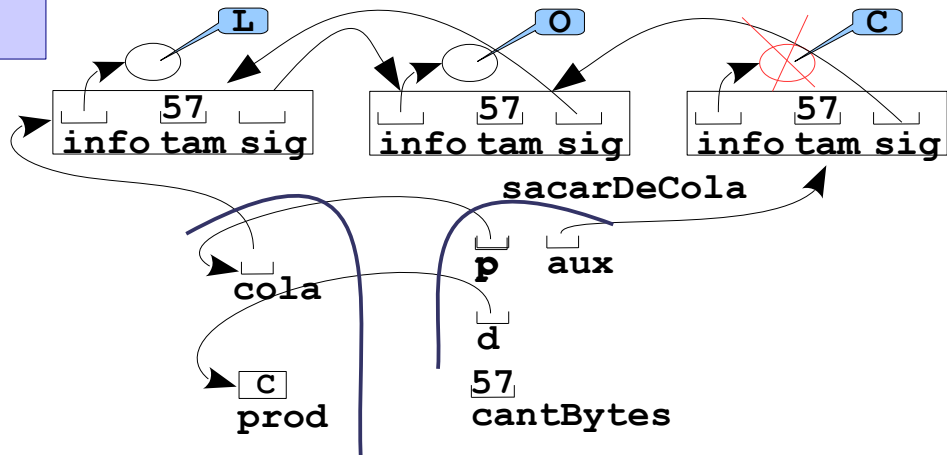


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

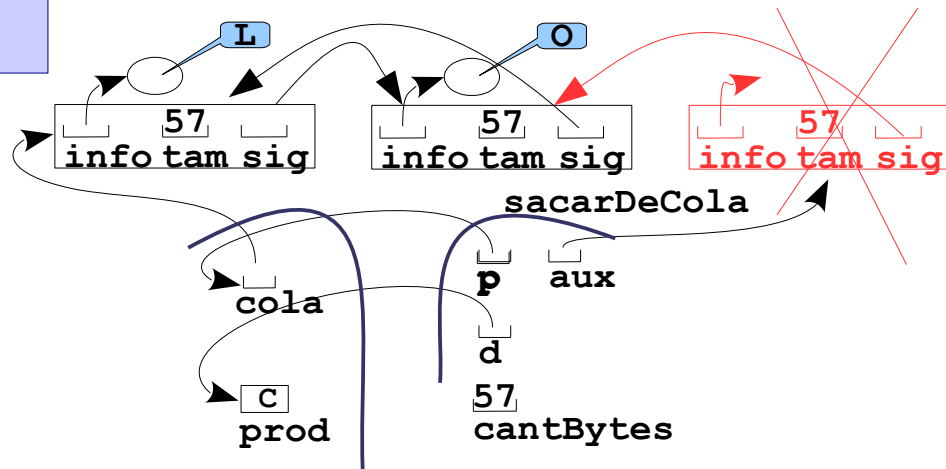


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

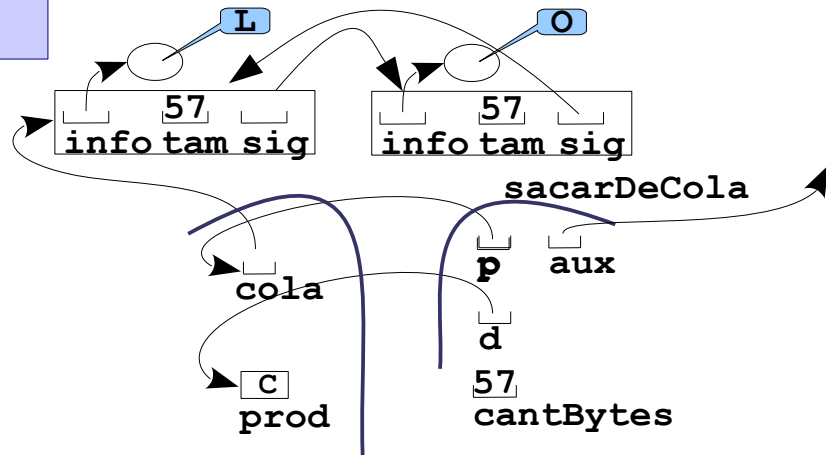


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

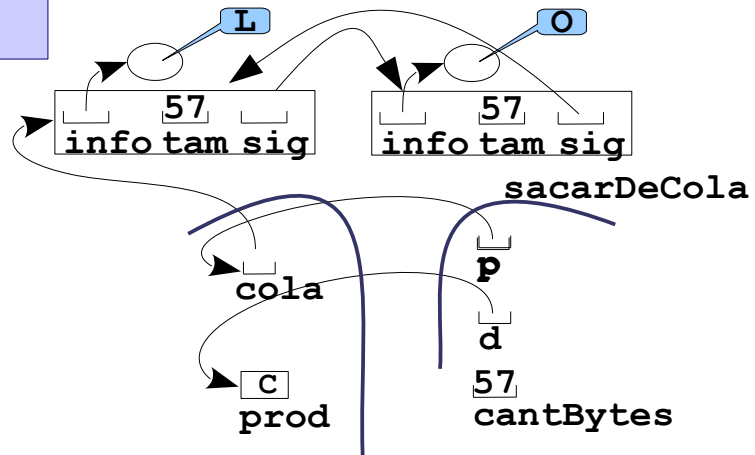


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

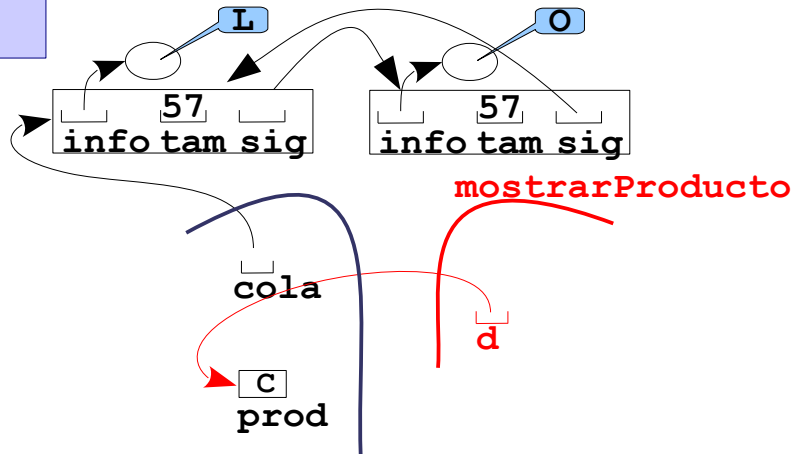


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(col, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

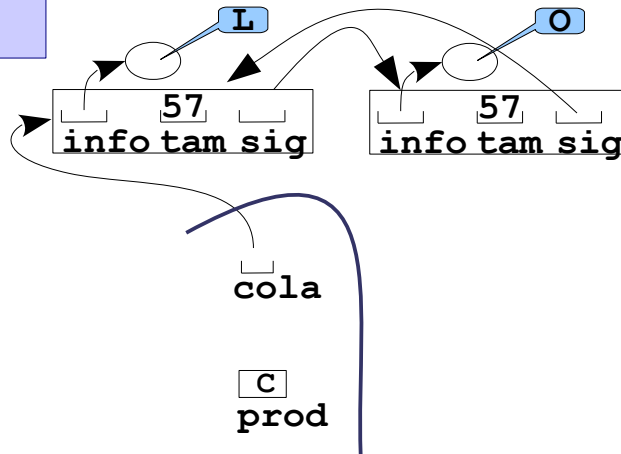


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(colas, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



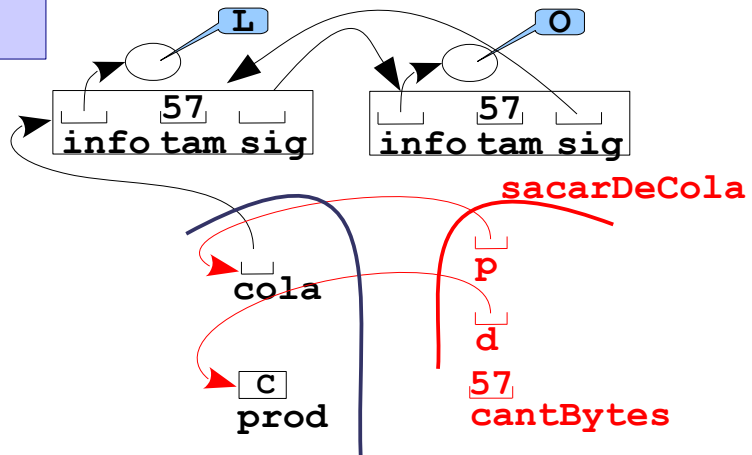
```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

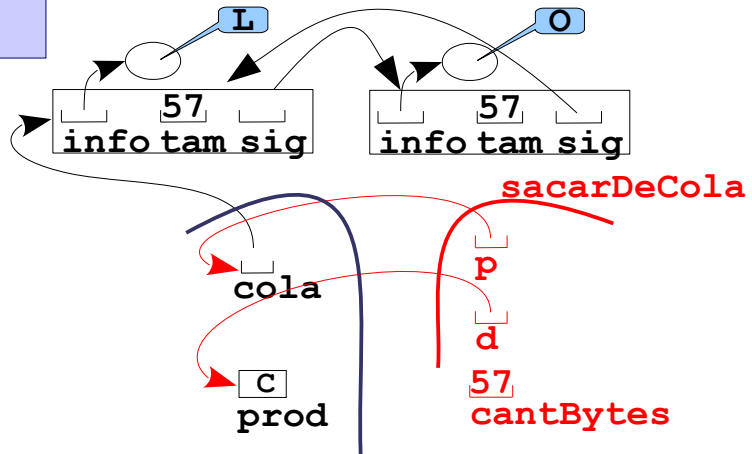


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(col, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

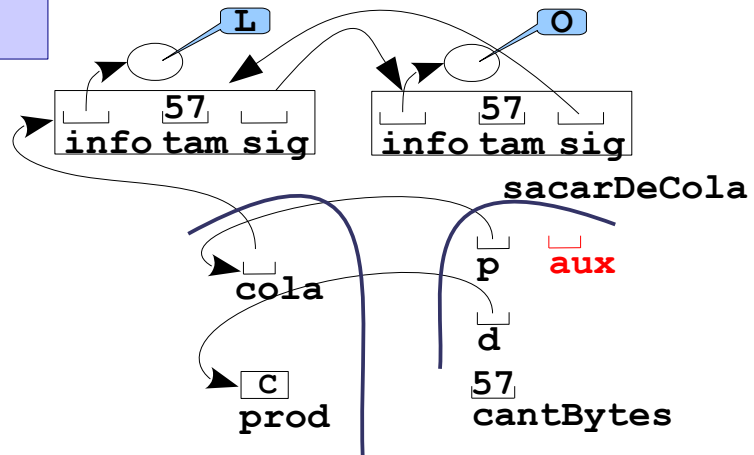


```
cola.c x cola.c x cola.h x *main.c x main.h x
101 59 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102 60 {
103 61     tNodo *aux;
104 62
105 63     if(*p == NULL)
106 64         return 0;
107 65     aux = (*p)->sig;
108 66     memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109 67     if(aux == *p)
110 68         *p = NULL;
111 69     else
112 70         (*p)->sig = aux->sig;
113 71     free(aux->info);
114 72     free(aux);
115 73     return 1;
116 74 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

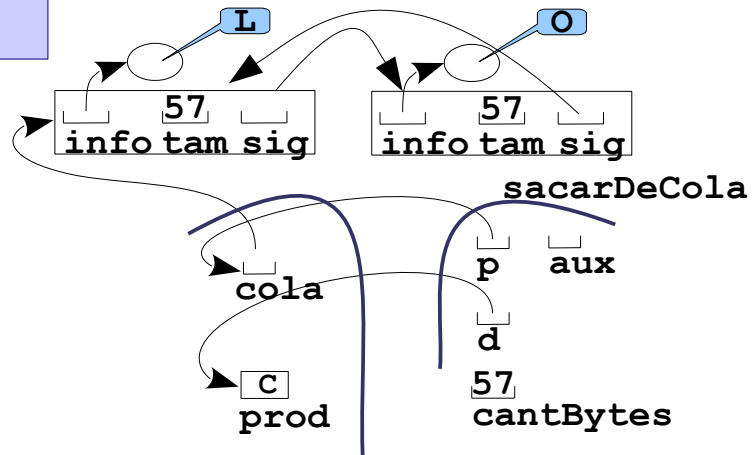


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61      tNodo *aux;
104      62
105      63      if(*p == NULL)
106      64      return 0;
107      65      aux = (*p)->sig;
108      66      memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67      if(aux == *p)
110      68      *p = NULL;
111      69      else
112      70      (*p)->sig = aux->sig;
113      71      free(aux->info);
114      72      free(aux);
115      73      return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

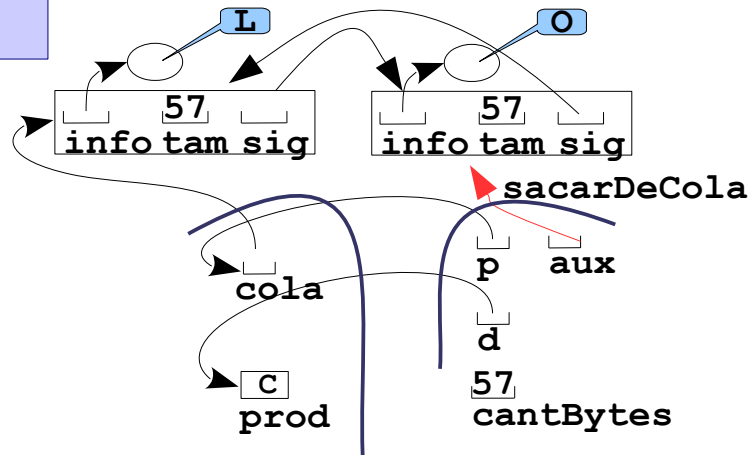


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

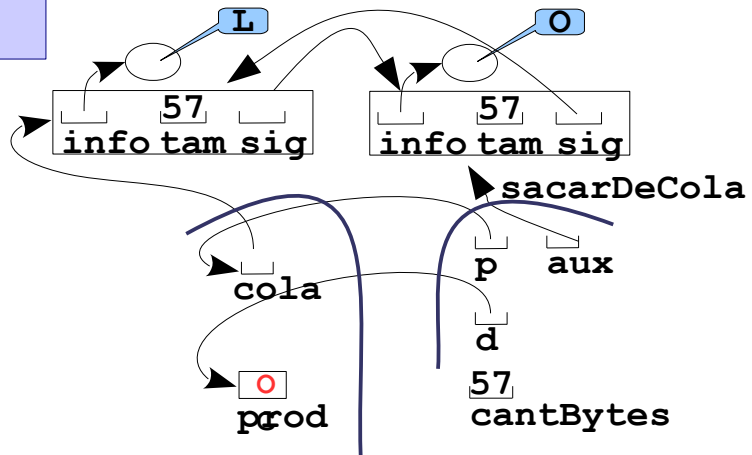


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

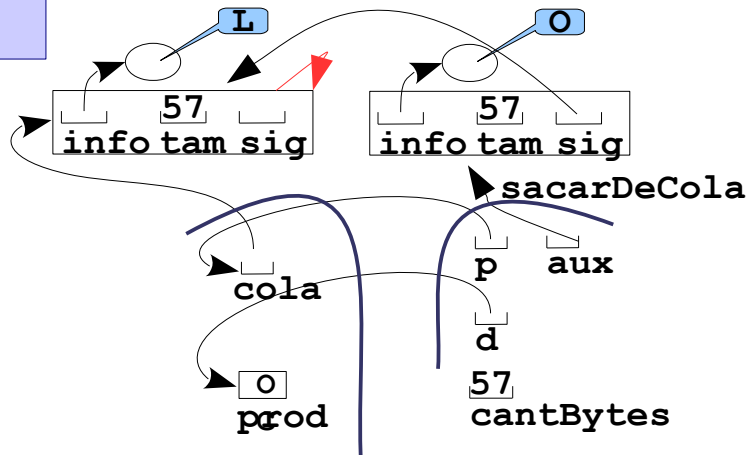


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

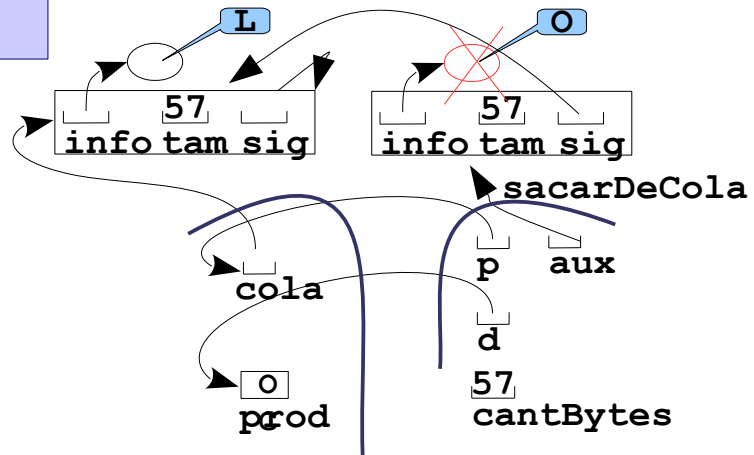


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



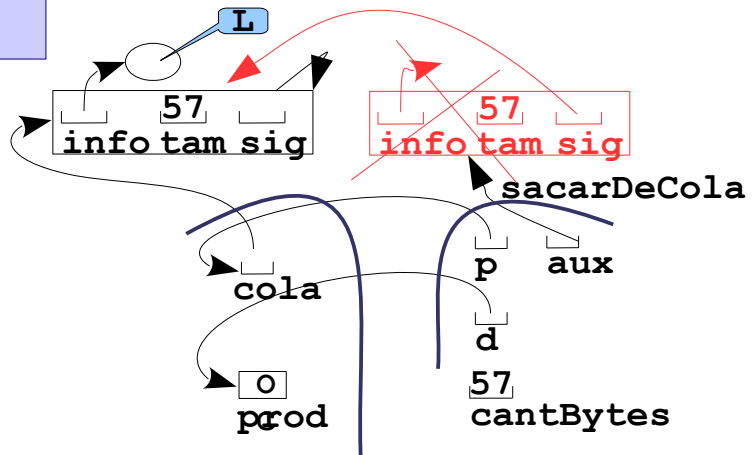
```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

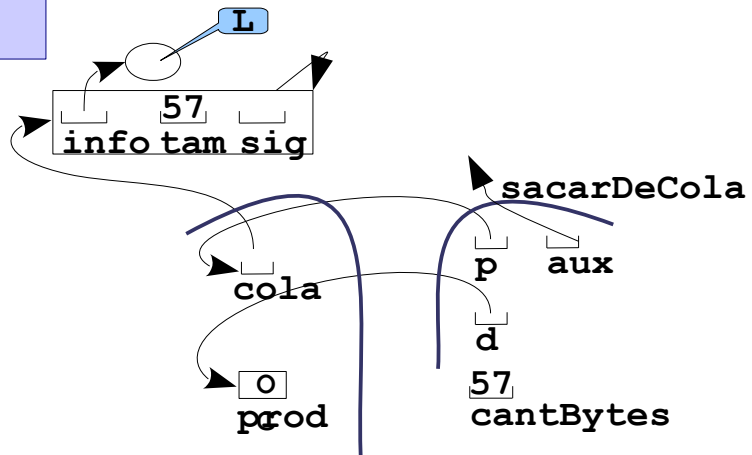


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

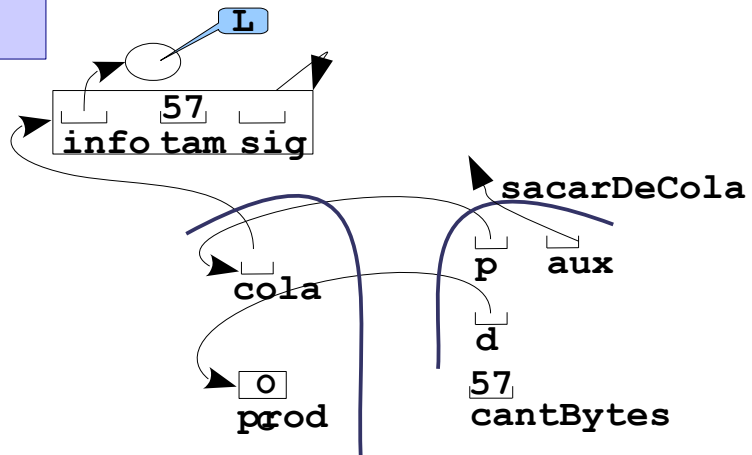


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

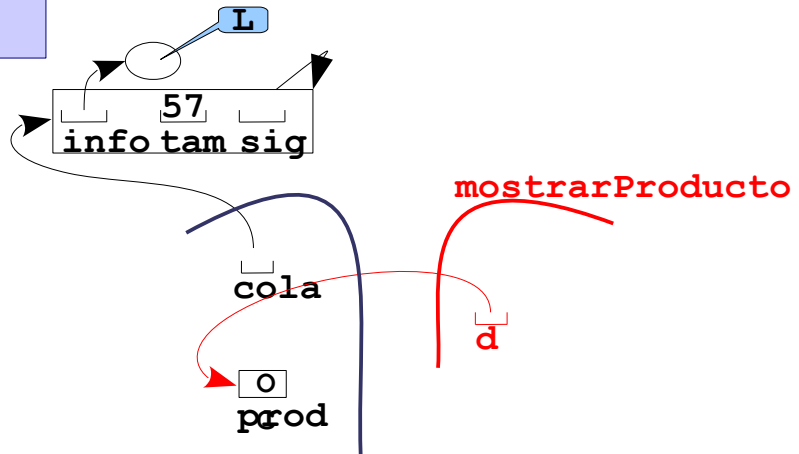


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(col, &prod, sizeof(prod))) {
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

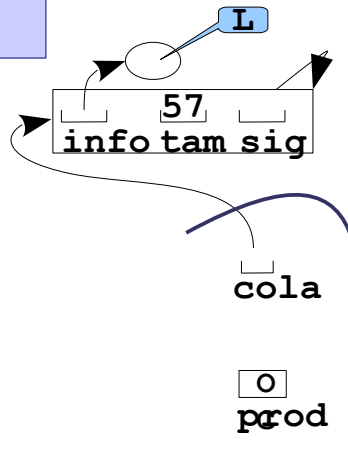


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(colas, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

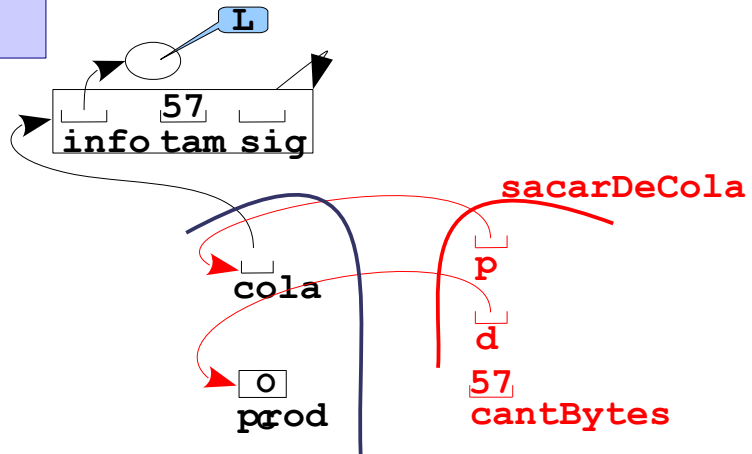


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

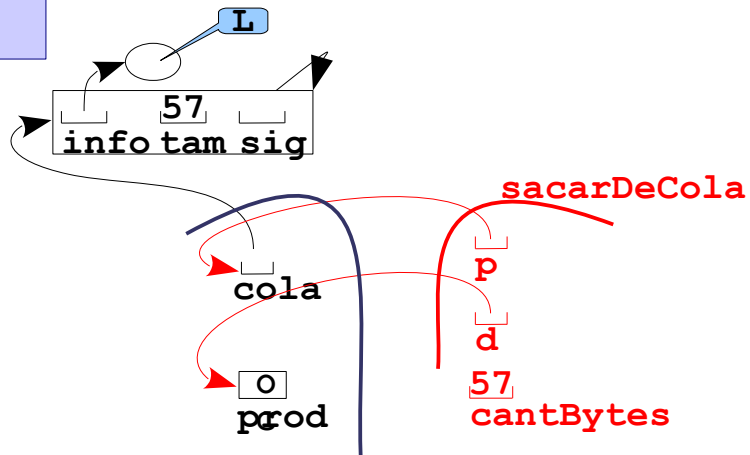


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(col, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

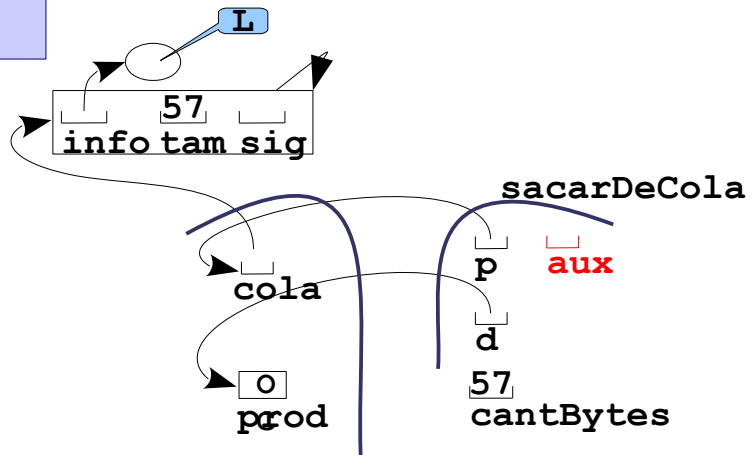


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



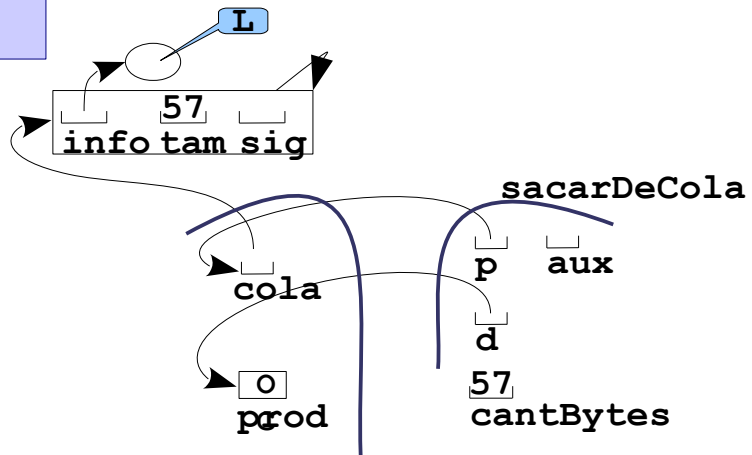
```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

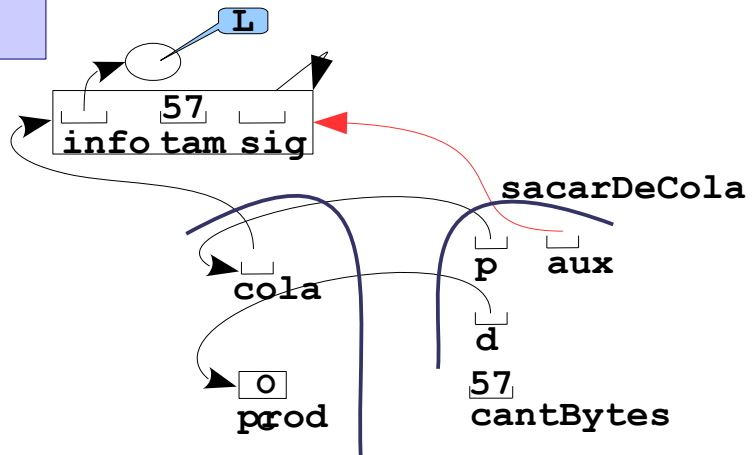


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

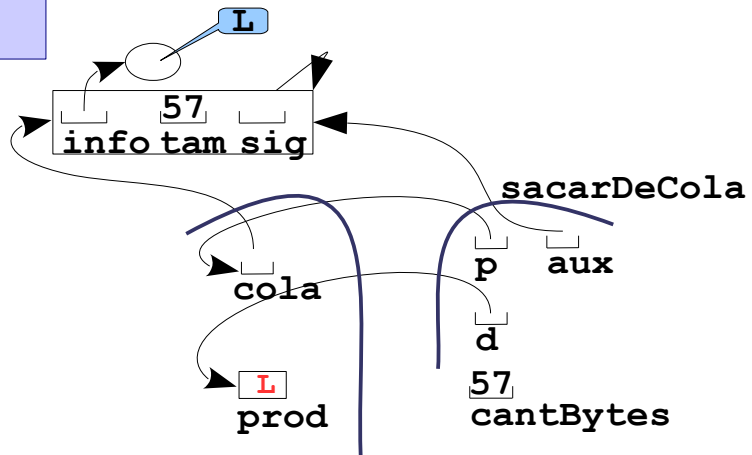


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

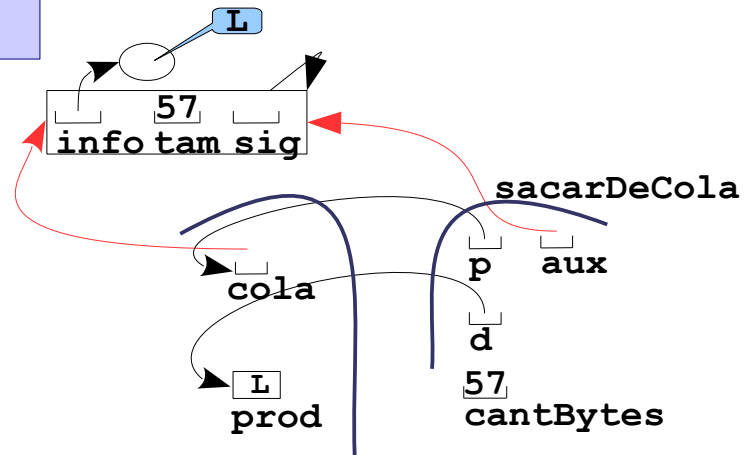


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

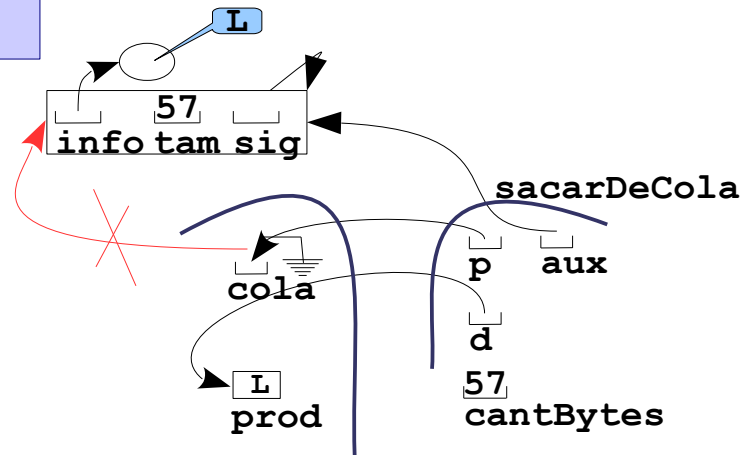


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.

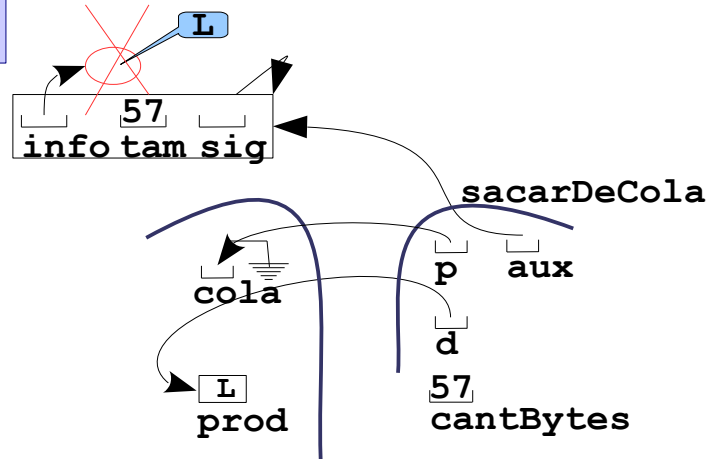


```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

## Tipo de dato Cola en Lista Enlazada Circular.

### Implementación Dinámica.



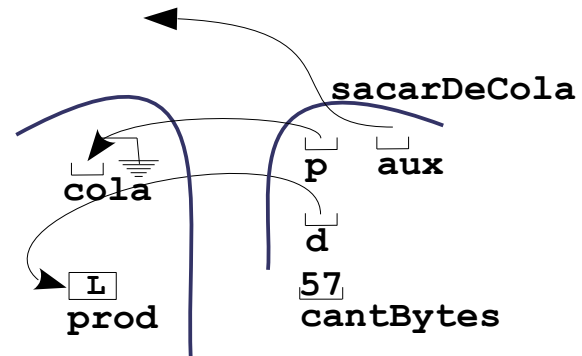
```
cola.c x cola.c x cola.h x *main.c x main.h x
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



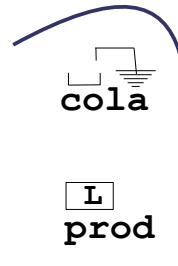
```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int  sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```



# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

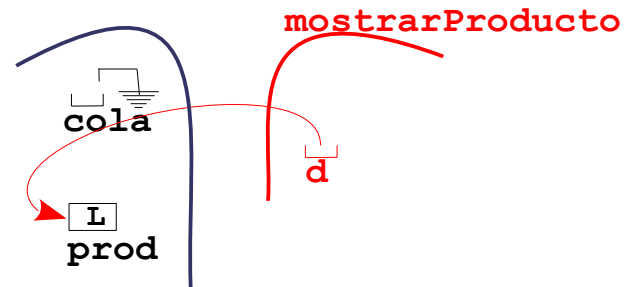


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))!
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

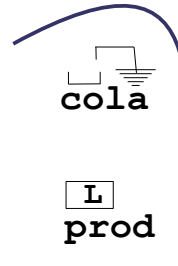


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(colas, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

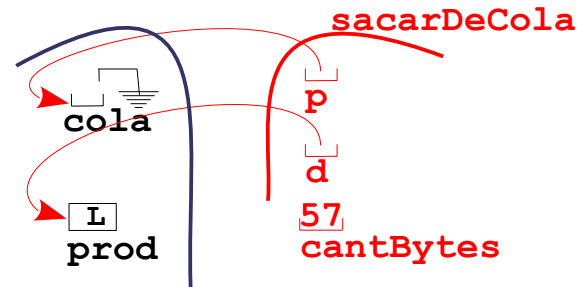


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))
102 {
103     canti--;
104     [mostrarProducto(&prod);]
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

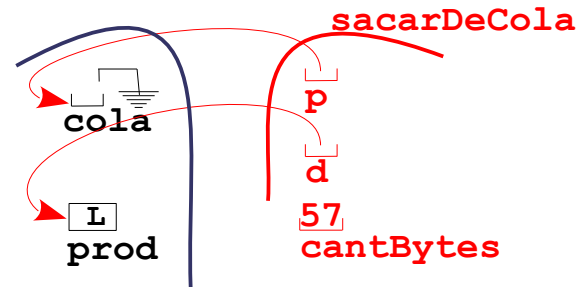


```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

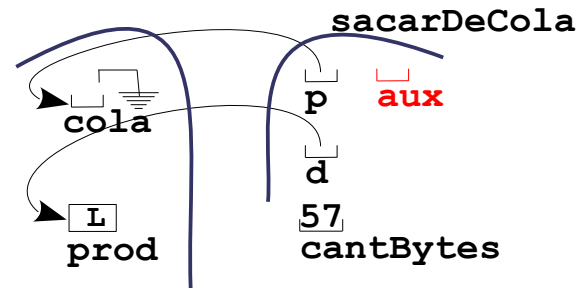


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

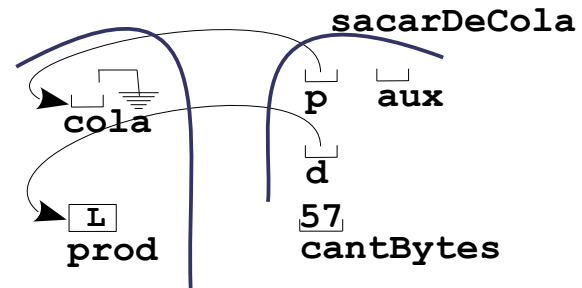


```
cola.c X cola.c X cola.h X *main.c X main.h X
101      59      int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
102      60      {
103      61          tNodo *aux;
104      62
105      63          if(*p == NULL)
106      64              return 0;
107      65          aux = (*p)->sig;
108      66          memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
109      67          if(aux == *p)
110      68              *p = NULL;
111      69          else
112      70              (*p)->sig = aux->sig;
113      71          free(aux->info);
114      72          free(aux);
115      73          return 1;
116      74      }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.

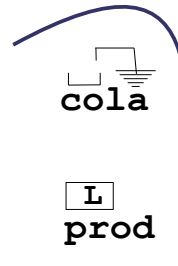


```
cola.c X cola.c X cola.h X *main.c X main.h X
59 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
60 {
61     tNodo *aux;
62
63     if(*p == NULL)
64         return 0;
65     aux = (*p)->sig;
66     memcpy(d, aux->info, minimo(cantBytes, aux->tamInfo));
67     if(aux == *p)
68         *p = NULL;
69     else
70         (*p)->sig = aux->sig;
71     free(aux->info);
72     free(aux);
73     return 1;
74 }
```

# Tipos de Datos Abstractos

Tipo de dato Cola en Lista Enlazada Circular.

Implementación Dinámica.



```
cola.c X cola.h X *main.c X main.h X
101 while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))
102 {
103     canti--;
104     mostrarProducto(&prod);
105 }
```



