

# Ramirez Manriquez Luis Fernando

## Capturas

```
main.c
98
99 int main(){
100     struct Nodo *inicio = NULL;
101     int n, valor, borrar;
102
103     printf("Cuantos valores insertar al inicio? ");
104     scanf("%d", &n);
105     for(int i=0; i<n; i++){
106         printf("Valor: ");
107         scanf("%d", &valor);
108         insertarInicio(&inicio, valor);
109     }
110     recorrerAdelante(inicio);
111
112     printf("\nCuantos valores insertar al final? ");
113     scanf("%d", &n);
114     for(int i=0; i<n; i++){
115         printf("Valor: ");
116         scanf("%d", &valor);
117         insertarFinal(&inicio, valor);
118     }
119     recorrerAdelante(inicio);
120
121
Cuantos valores insertar al inicio? 3
Valor: 4
Valor: 7
Valor: 3

Lista hacia adelante: 3 7 4

Cuantos valores insertar al final? 2
Valor: 8
Valor: 2

Lista hacia adelante: 3 7 4 8 2
Lista hacia atras: 2 8 4 7 3

Valor a eliminar: 8
Elemento eliminado correctamente.

Lista hacia adelante: 3 7 4 2
Memoria liberada.

...Program finished with exit code 0
Press ENTER to exit console.
```

The screenshot shows a development environment with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and project management. Below the toolbar, the code editor displays a file named "main.c". The code implements a circular linked list. It first asks for the number of values to insert (3 in this case). It then inserts three values (6, 4, 9) into the list. After insertion, it prints the list (6 4 9). It then asks for a value to delete (6), removes it from the list, and prints the updated list (4 9). Finally, it prints a message indicating the program has finished.

```
main.c
70     struct Nodo *inicio = NULL;
71     int n, valor, borrar;
72
73     printf("Cuantos valores insertar? ");
74     scanf("%d", &n);
75
76     for(int i=0; i<n; i++){
77         printf("Valor: ");
78         scanf("%d", &valor);
79         insertarCircular(&inicio, valor);
80     }
81
82     recorrerCircular(inicio);
83
84     printf("\nValor a eliminar: ");
85     scanf("%d", &borrar);
86     eliminarCircular(&inicio, borrar);
87
88     recorrerCircular(inicio);
89
90     return 0;
91 }
92 
```

Cuantos valores insertar? 3  
Valor: 6  
Valor: 4  
Valor: 9  
Lista circular: 6 4 9  
Valor a eliminar: 6  
Lista circular: 4 9  
...Program finished with exit code 0  
Press ENTER to exit console.

## Actividad

### Preguntas (A)

1. ¿Por qué es más fácil eliminar un nodo intermedio aquí?

Porque el nodo ya tiene guardados quién va antes y quién va después, así no tienes que andar buscando mucho, solo lo desconectas y ya

2. ¿Qué pasa si olvidas actualizar los punteros?

La lista se desacomoda, se rompe y ya nada enlaza bien, el programa puede fallar o ni dejarte recorrerla

3. ¿Dónde se usan en la vida real?

En cosas donde puedes ir adelante y atrás, como el historial del navegador, playlist de música o menús interactivos.

### Preguntas (B)

1. ¿Diferencia entre doble ligada y circular doble?

La normal termina, la circular vuelve al inicio y nunca se acaba, es un ciclo

2. ¿Qué error causa ciclos infinitos?

Si no controlas cuándo parar, la lista da vueltas sin fin y se traba el programa

3. ¿En qué se usa en la vida real?

En playlists repetitivas, menús que giran, juegos por turnos, sliders de imágenes.