

GRUPO 9 – Correcciones Trabajo Práctico: COLAS

```
04-trabajoPractico-Colas > C main.c > main()
12 int main()
25 while(ejercicio != -1){
27     switch (ejercicio)
32     while(subejercicio != -1){
34         switch (subejercicio)
55         printf("Nueva cola: \n");
56         c_mostrar(c_resultado);
57         c_destruir(c_resultado);
58         break;
59         case 4: printf("d. Contar los elementos de la cola.\n");
60                 valor = c_ej2_contarelementos(c);
61                 printf("La cantidad de elementos es: %d \n", valor);
62                 break;
63         case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64                 c_resultado = c_ej2_copiar(c);
65                 printf("Copia de la cola: \n");
66                 c_mostrar(c_resultado);
67                 printf("Cola original: \n");
68                 c_mostrar(c);
```

```
Complejidad algoritmica O(n) = n^2 (Cuadratica).
Ingrese el ejercicio (2-7) o -1 para salir:
6
Ejercicio 6
Cargue la pila.
Ingrese el tamaño de la pila: 1
Ingrese el valor: 1
Cargue la cola.
Ingrese el tamaño de la cola: 1
Ingrese el valor: 6
Clave : Pila : Cola
1 : 1 : 2
Clave : Pila : Cola
2 : 2 : 1
Clave : Pila : Cola
1 : 3 : 2
Complejidad algoritmica O(n) = n^2 (Cuadratica).
Ingrese el ejercicio (2-7) o -1 para salir:
```

Ejercicio 5: permite cargar ceros y tira un exception. No controla que los datos ingresados sean mayores o iguales a 2. Muestra información confusa por pantalla y muchas veces incorrecta.

```
04-trabajoPractico-Colas > C tp4.c > c_ej5_divisortotal(Cola c){
219 Cola c_ej5_divisortotal(Cola c){
221     int Max = 0, Min, contador_actual, mitad, resto, cantidad;
228     bool salir = false;
229     cantidad = c_ej2_contarelementos(c);
230     //Obtengo el %50 y redondeo para arriba
231     mitad = (cantidad % 2 == 0) ? cantidad / 2 : (cantidad / 2) + 1;
232     //-----Buscar el MAX y MIN y copio en ColaAux para no perder la cola-----
233     aux = c_recuperar(c);
234     Min = aux->clave;
235     while(!c_es_vacia(c)){
236         aux = c_desencolar(c);
237         if(aux->clave < Min){
238             Min = aux->clave;
239         }
240         if(aux->clave > Max){
241             Max = aux->clave;
```

```
Exception has occurred.
Segmentation fault
```

```
Ingrese el ejercicio (2-7) o -1 para salir:
5
Ejercicio 5
Cargue la cola.
Ingrese el tamaño de la cola: 0
```

GRUPO 9 – Correcciones Trabajo Práctico: COLAS

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25     while(ejercicio != -1){
27         switch (ejercicio)
32             while(subejercicio != -1){
34                 switch (subejercicio)
55                     printf("Nueva cola: \n");
56                     c_mostrar(c_resultado);
57                     c_destruir(c_resultado);
58                     break;
59                     case 4: printf("d. Contar los elementos de la cola.\n");
60                         valor = c_ej2_contarelementos(c);
61                         printf("La cantidad de elementos es: %d \n", valor);
62                         break;
63                     case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64                         c_resultado = c_ej2_copiar(c);
65                         printf("Copia de la cola: \n");
66                         c_mostrar(c_resultado);
67                         printf("Cola original: \n");
68                         c_mostrar(c);
```

OUTPUT TERMINAL PORTS

cppdbg: main

Ingrese el ejercicio (2-7) o -1 para salir:
5
Ejercicio 5
Cargue la cola.
Ingrese el tamaño de la cola: 1
Ingrese el valor: 1
Valor Total 1
Complejidad algoritmica $O(n) = n^2$ (Cuadratica).
Ingrese el ejercicio (2-7) o -1 para salir:

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25     while(ejercicio != -1){
27         switch (ejercicio)
32             while(subejercicio != -1){
34                 switch (subejercicio)
55                     printf("Nueva cola: \n");
56                     c_mostrar(c_resultado);
57                     c_destruir(c_resultado);
58                     break;
59                     case 4: printf("d. Contar los elementos de la cola.\n");
60                         valor = c_ej2_contarelementos(c);
61                         printf("La cantidad de elementos es: %d \n", valor);
62                         break;
63                     case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64                         c_resultado = c_ej2_copiar(c);
65                         printf("Copia de la cola: \n");
66                         c_mostrar(c_resultado);
67                         printf("Cola original: \n");
68                         c_mostrar(c);
```

OUTPUT TERMINAL PORTS

cppdbg: main

5
Ejercicio 5
Cargue la cola.
Ingrese el tamaño de la cola: 1
Ingrese el valor: 1
Valor Total 1
Complejidad algoritmica $O(n) = n^2$ (Cuadratica).
Ingrese el ejercicio (2-7) o -1 para salir:
5
Ejercicio 5
Cargue la cola.
Ingrese el tamaño de la cola: 2
Ingrese el valor: 2
Ingrese el valor: 4
Valor Total 2
Complejidad algoritmica $O(n) = n^2$ (Cuadratica).
Ingrese el ejercicio (2-7) o -1 para salir:

GRUPO 9 – Correcciones Trabajo Práctico: COLAS

Visual Studio Code interface showing the implementation of a queue in C. The code is in `main.c` and the terminal shows the execution results.

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25 while(ejercicio != -1){
27     switch (ejercicio)
32     while(subejercicio != -1){
34         switch (subejercicio)
55         printf("Nueva cola: \n");
56         c_mostrar(c_resultado);
57         c_destruir(c_resultado);
58         break;
59         case 4: printf("d. Contar los elementos de la cola.\n");
60                 valor = c_ej2_contarelementos(c);
61                 printf("La cantidad de elementos es: %d \n", valor);
62                 break;
63         case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64                 c_resultado = c_ej2_copiar(c);
65                 printf("Copia de la cola: \n");
66                 c_mostrar(c_resultado);
67                 printf("Cola original: \n");
68                 c_mostrar(c);
```

Terminal output:

```
Ingresa el valor: 4
Valor/es Parciales 2
Valor Total 1
Complejidad algoritmica O(n) = n^2 (Cuadratica).
Ingresa el ejercicio (2-7) o -1 para salir:
5
Ejercicio 5
Cargue la cola.
Ingresa el tamaño de la cola: 4
Ingresa el valor: 2
Ingresa el valor: 4
Ingresa el valor: 8
Ingresa el valor: 12
Valor/es Parciales 4
Valor/es Parciales 2
Valor Total 1
Complejidad algoritmica O(n) = n^2 (Cuadratica).
Ingresa el ejercicio (2-7) o -1 para salir:
```

Visual Studio Code interface showing the implementation of a queue in C. The code is in `main.c` and the terminal shows the execution results.

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25 while(ejercicio != -1){
27     switch (ejercicio)
32     while(subejercicio != -1){
34         switch (subejercicio)
55         printf("Nueva cola: \n");
56         c_mostrar(c_resultado);
57         c_destruir(c_resultado);
58         break;
59         case 4: printf("d. Contar los elementos de la cola.\n");
60                 valor = c_ej2_contarelementos(c);
61                 printf("La cantidad de elementos es: %d \n", valor);
62                 break;
63         case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64                 c_resultado = c_ej2_copiar(c);
65                 printf("Copia de la cola: \n");
66                 c_mostrar(c_resultado);
67                 printf("Cola original: \n");
68                 c_mostrar(c);
```

Terminal output:

```
Ingresa el valor: 8
Ingresa el valor: 12
Valor/es Parciales 4
Valor/es Parciales 2
Valor Total 1
Complejidad algoritmica O(n) = n^2 (Cuadratica).
Ingresa el ejercicio (2-7) o -1 para salir:
5
Ejercicio 5
Cargue la cola.
Ingresa el tamaño de la cola: 2
Ingresa el valor: 2
Ingresa el valor: 4
Valor/es Parciales 4
Valor/es Parciales 2
Valor Total 1
Complejidad algoritmica O(n) = n^2 (Cuadratica).
Ingresa el ejercicio (2-7) o -1 para salir:
```

GRUPO 9 – Correcciones Trabajo Práctico: COLAS

Ejercicio 6 no muestra el contenido de las estructuras luego de ejecutado el ejercicio.

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25 while(ejercicio != -1){
29     case 2: printf("Ejercicio 2\n");
53
54     valor = pedir numero("Ingrese el elemento: \n");
55     c_resultado = c.ej2_sacarelemento(c, valor);
56     printf("Nueva cola: \n");
57     c_mostrar(c_resultado);
58     c_destruir(c_resultado);
59     break;
60     case 4: printf("d. Contar los elementos de la cola.\n");
```

```
OUTPUT TERMINAL PORTS
c_cppdbg: main
Ingrese el ejercicio (2-7) o -1 para salir:
6
Ejercicio 6
Cargue la pila.
Ingrese el tamaño de la pila: 2
Ingrese el valor: 1
Ingrese el valor: 2
Cargue la cola.
Ingrese el tamaño de la cola: 2
Ingrese el valor: 1
Ingrese el valor: 5
Clave : Pila : Cola
1 : 1 : 1
Complejidad algorítmica 0(n) = n^2 (Cuadrática).
Ingrese el ejercicio (2-7) o -1 para salir:
```

Ejercicio 7: Permite cargar cero en tiempo de atención, con lo cual, entra en un loop infinito. Permite cargar cero como tiempo en las colas. Solamente funciona con el ejemplo de la guía

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25 while(ejercicio != -1){
27     switch (ejercicio)
32     while(subejercicio != -1){
34         switch (subejercicio)
55         printf("Nueva cola: \n");
56         c_mostrar(c_resultado);
57         c_destruir(c_resultado);
58         break;
59         case 4: printf("d. Contar los elementos de la cola.\n");
60         valor = c.ej2_contarelementos(c);
61         printf("La cantidad de elementos es: %d \n", valor);
62         break;
63         case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64         c_resultado = c.ej2_copiar(c);
65         printf("Copia de la cola: \n");
66         c_mostrar(c_resultado);
67         printf("Cola original: \n");
68         c_mostrar(c);
```

```
OUTPUT TERMINAL PORTS
c_cppdbg: main
Clave : Pila : Cola
1 : 1 : 2
Clave : Pila : Cola
2 : 2 : 1
Clave : Pila : Cola
1 : 3 : 2
Complejidad algorítmica 0(n) = n^2 (Cuadrática).
Ingrese el ejercicio (2-7) o -1 para salir:
7
Ejercicio 7
Cargue la cola 1.
Ingrese el tamaño de la cola: 0
Cargue la cola 2.
Ingrese el tamaño de la cola: 0
Cargue la cola 3.
Ingrese el tamaño de la cola: 0
Ingresar el tiempo de atención.
0
```

GRUPO 9 – Correcciones Trabajo Práctico: COLAS

The screenshot shows the Visual Studio Code editor with a C program for queue operations. The code is in `main.c` and includes a `main` function that uses a `switch` statement to handle different exercises. The terminal output shows the program's execution, including user input and the resulting queue state.

```
04-trabajoPractico-Colas > C main.c > main()
12 int main(){
25     while(ejercicio != -1){
27         switch (ejercicio)
32             while(subejercicio != -1){
34                 switch (subejercicio)
55                     printf("Nueva cola: \n");
56                     c_mostrar(c_resultado);
57                     c_destruir(c_resultado);
58                     break;
59                 case 4: printf("d. Contar los elementos de la cola.\n");
60                     valor = c_ej2_contarelementos(c);
61                     printf("La cantidad de elementos es: %d \n", valor);
62                     break;
63                 case 5: printf("e. Realizar una funcion que realice una copia de una cola\n");
64                     c_resultado = c_ej2_copiar(c);
65                     printf("Copia de la cola: \n");
66                     c_mostrar(c_resultado);
67                     printf("Cola original: \n");
68                     c_mostrar(c);
```

OUTPUT TERMINAL PORTS

cpdbg: main

```
> > > TERMINAL
Ingrese el valor: 5
Ingrese el valor: 3
Cargue la cola 3.
Ingrese el tamaño de la cola: 1
Ingrese el valor: 10
Ingrese el tiempo de atencion:
5
Cliente 1 || Cola 1
Cliente 1 || Cola 2
Cliente 2 || Cola 1
Cliente 2 || Cola 2
Cliente 1 || Cola 3
Cliente 3 || Cola 1
Cliente 3 || Cola 2
Cliente 4 || Cola 1
Cliente 4 || Cola 2
Complejidad algoritmica O(n) = n (Lineal).
Ingrese el ejercicio (2-7) o -1 para salir:
```