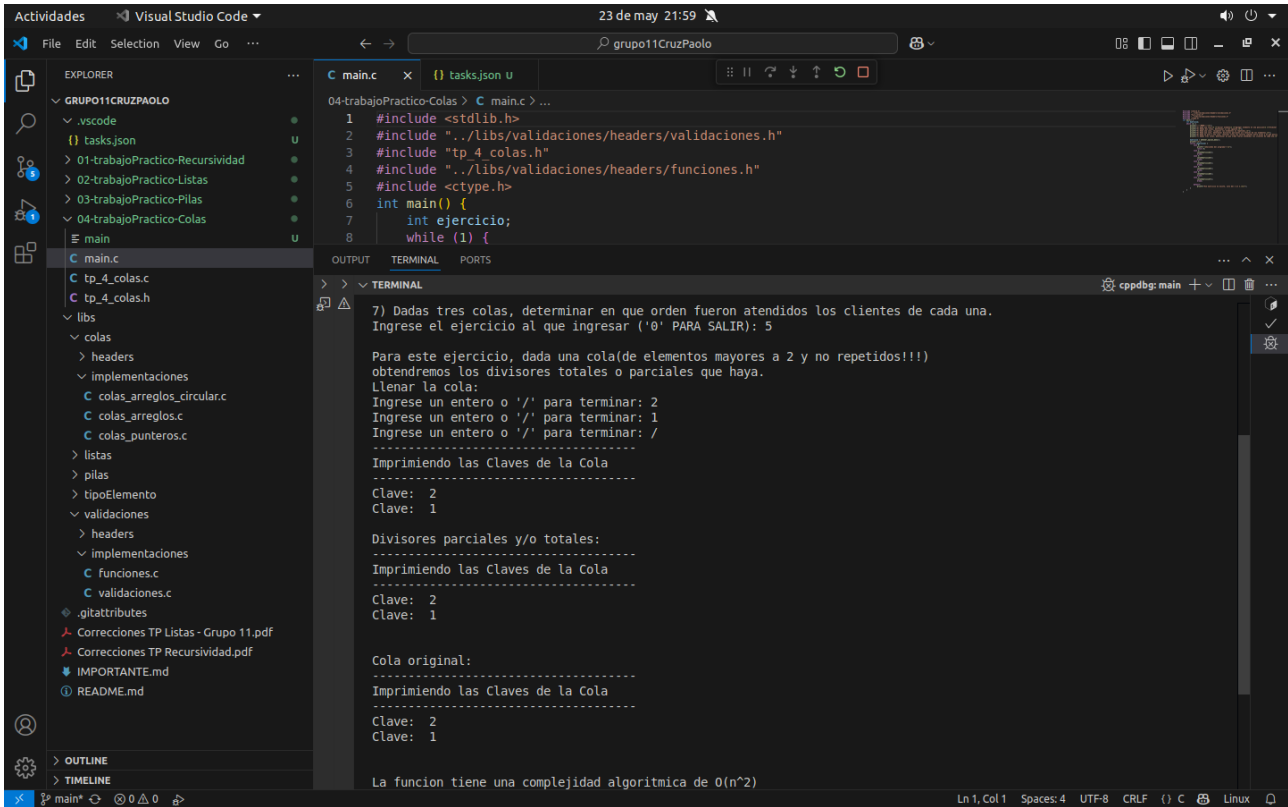


RESULTADO DE LA CORRECCIÓN: **DESAPROBADO**

OBSERVACIONES

Ejercicio 5: indica por pantalla que los valores ingresados deber ser mayores o iguales a 2, pero permite la carga de valores menores. Muestra en la misma cola los divisores totales y parciales sin identificar cuál es cual.



```
04-trabajoPractico-Colas > C main.c > ...
1 #include <stdlib.h>
2 #include "../libs/validaciones/headers/validaciones.h"
3 #include "tp_4_colas.h"
4 #include "../libs/validaciones/headers/funciones.h"
5 #include <ctype.h>
6 int main() {
7     int ejercicio;
8     while (1) {
9         // ...
10    }
11 }
```

7) Dadas tres colas, determinar en que orden fueron atendidos los clientes de cada una.
Ingrese el ejercicio al que ingresar ('0' PARA SALIR): 5

Para este ejercicio, dada una cola(de elementos mayores a 2 y no repetidos!!!)
obtendremos los divisores totales o parciales que haya.

Llenar la cola:
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: /

Imprimiendo las Claves de la Cola

Clave: 2
Clave: 1

Divisores parciales y/o totales:

Imprimiendo las Claves de la Cola

Clave: 2
Clave: 1

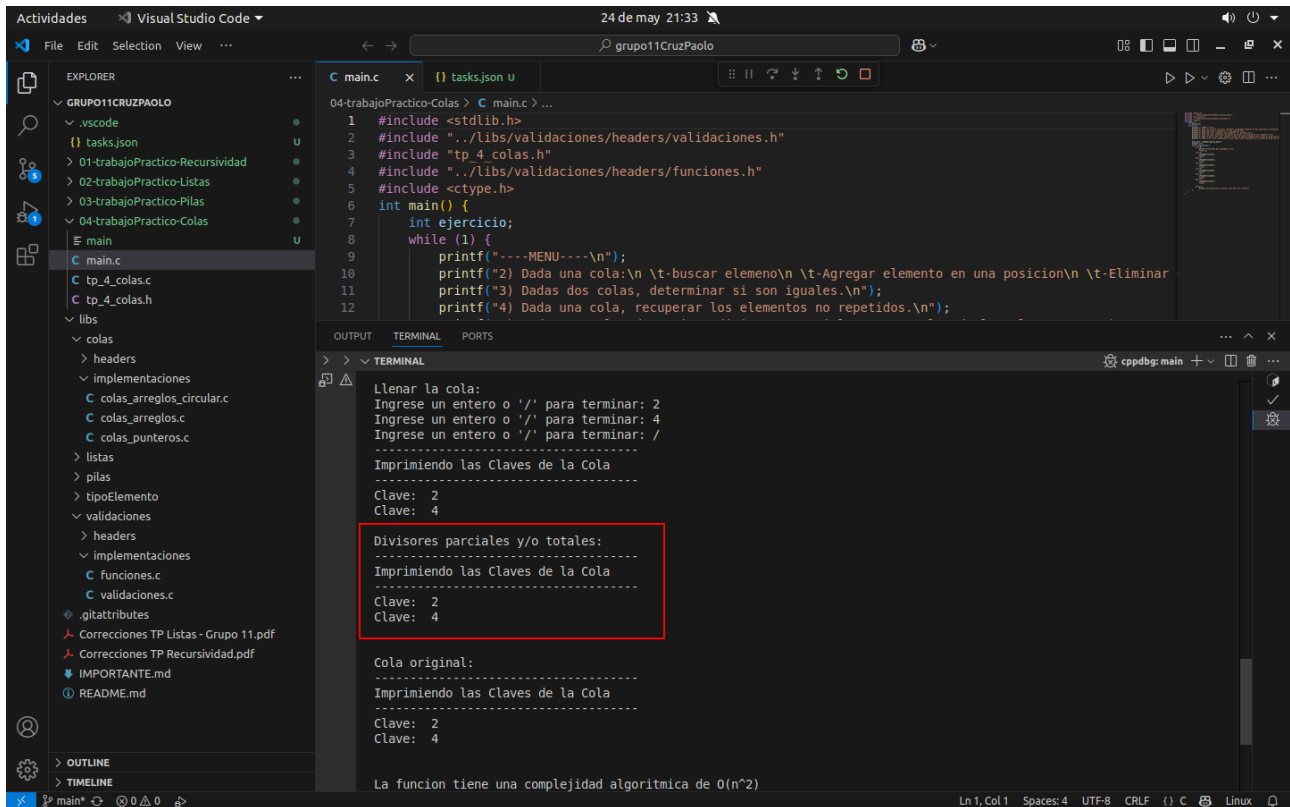
Cola original:

Imprimiendo las Claves de la Cola

Clave: 2
Clave: 1

La funcion tiene una complejidad algoritmica de $O(n^2)$

GRUPO 11 – Correcciones Trabajo Práctico: COLAS



```
04-trabajoPractico-Colas > C main.c > ...
1 #include <stdlib.h>
2 #include "../libs/validaciones/headers/validaciones.h"
3 #include "tp_4_colas.h"
4 #include "../libs/validaciones/headers/funciones.h"
5 #include <ctype.h>
6 int main() {
7     int ejercicio;
8     while (1) {
9         printf("----MENU----\n");
10        printf("2) Dada una cola:\n \t-buscar elemeno\n \t-Agregar elemento en una posicion\n \t-Eliminar
11        printf("3) Dadas dos colas, determinar si son iguales.\n");
12        printf("4) Dada una cola, recuperar los elementos no repetidos.\n");
13    }
14 }
```

OUTPUT TERMINAL PORTS

cpplib: main

Llenar la cola:
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: 4
Ingrese un entero o '/' para terminar: /

Imprimiendo las Claves de la Cola
Clave: 2
Clave: 4

Divisores parciales y/o totales:

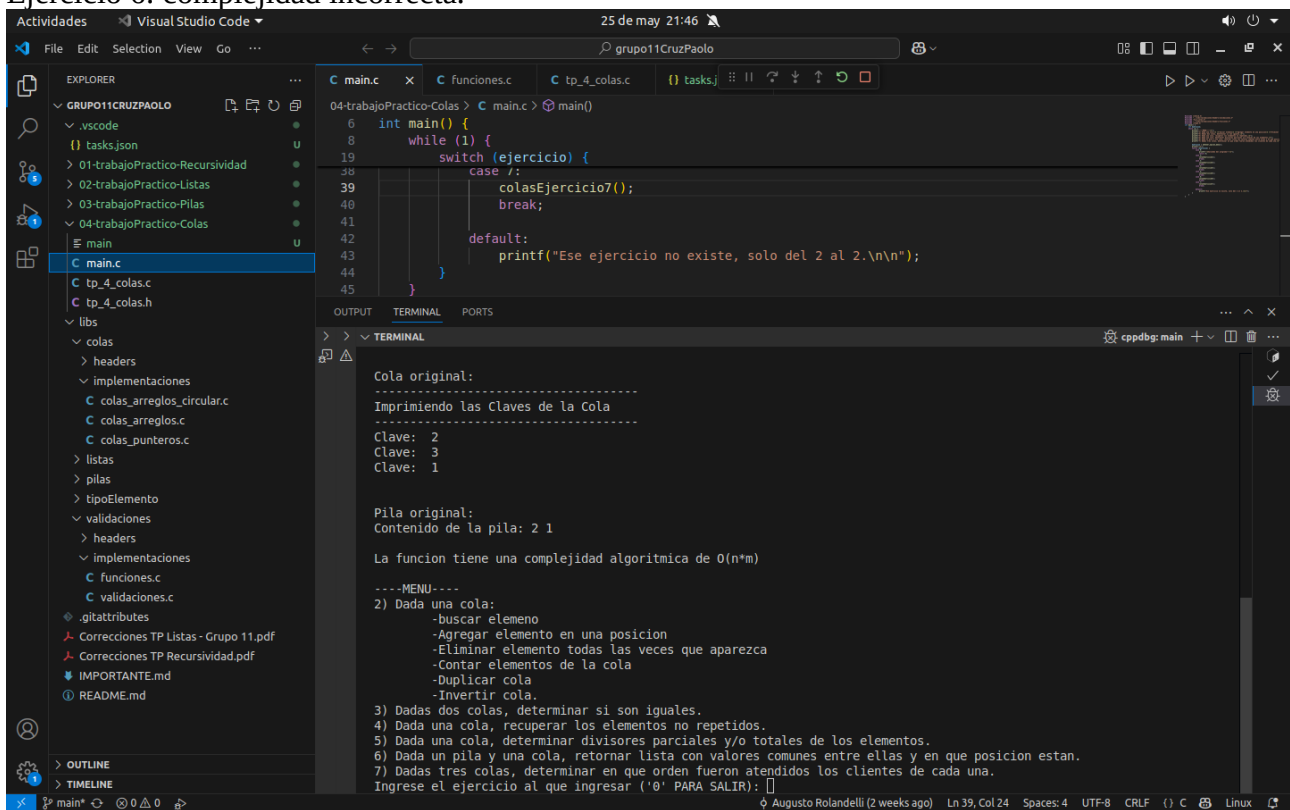
Imprimiendo las Claves de la Cola
Clave: 2
Clave: 4

Cola original:

Imprimiendo las Claves de la Cola
Clave: 2
Clave: 4

La funcion tiene una complejidad algoritmica de $O(n^2)$

Ejercicio 6: complejidad incorrecta.



```
04-trabajoPractico-Colas > C main.c > main()
6 int main() {
7     while (1) {
8         switch (ejercicio) {
9             case 1:
10                colasEjercicio7();
11                break;
12             default:
13                printf("Ese ejercicio no existe, solo del 2 al 2.\n\n");
14            }
15        }
16    }
```

OUTPUT TERMINAL PORTS

cpplib: main

Cola original:

Imprimiendo las Claves de la Cola
Clave: 2
Clave: 3
Clave: 1

Pila original:
Contenido de la pila: 2 1

La funcion tiene una complejidad algoritmica de $O(n^m)$

----MENU----

2) Dada una cola:
-buscar elemeno
-Agregar elemento en una posicion
-Eliminar elemento todas las veces que aparezca
-Contar elementos de la cola
-Duplicar cola
-Invertir cola

3) Dadas dos colas, determinar si son iguales.

4) Dada una cola, recuperar los elementos no repetidos.

5) Dada una cola, determinar divisores parciales y/o totales de los elementos.

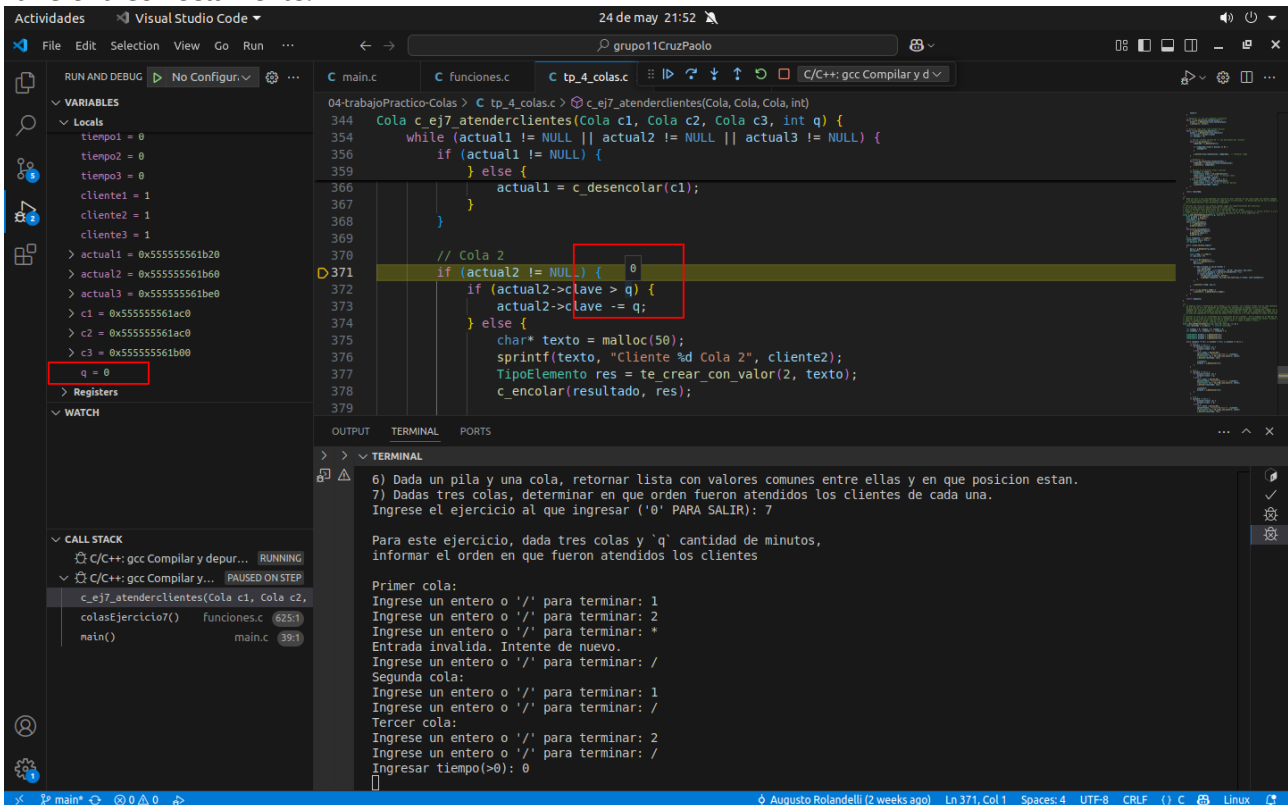
6) Dada una pila y una cola, retornar lista con valores comunes entre ellas y en que posicion estan.

7) Dadas tres colas, determinar en que orden fueron atendidos los clientes de cada una.

Ingrese el ejercicio al que ingresar ('0' PARA SALIR):

GRUPO 11 – Correcciones Trabajo Práctico: COLAS

Ejercicio 7: Permite cargar cero en tiempo de atención, con lo cual, entra en un loop infinito. No funciona correctamente.

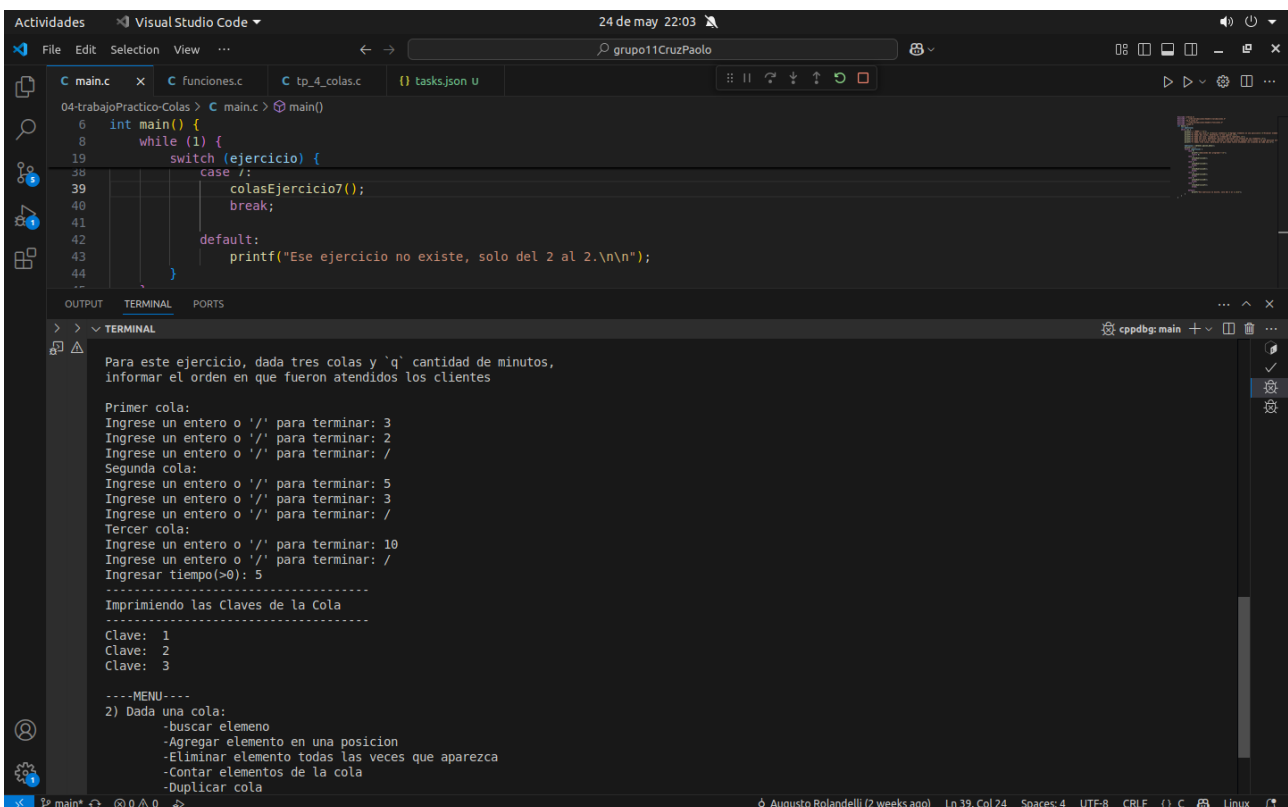


```
04-trabajoPractico-Colas > C tp_4_colas.c > c_ej7_atenderclientes(Cola c1, Cola c2, Cola c3, int q) {
344 Cola c_ej7_atenderclientes(Cola c1, Cola c2, Cola c3, int q) {
354 while (actual1 != NULL || actual2 != NULL || actual3 != NULL) {
356     if (actual1 != NULL) {
359         } else {
366             actual1 = c_desencolar(c1);
367         }
368     }
369 }
370 // Cola 2
371 if (actual2 != NULL) { 0
372     if (actual2->clave > q) {
373         actual2->clave -= q;
374     } else {
375         char* texto = malloc(50);
376         sprintf(texto, "Cliente %d Cola 2", cliente2);
377         TipoElemento res = te_crear_con_valor(2, texto);
378         c_encolar(resultado, res);
379     }
}
```

6) Dada un pila y una cola, retornar lista con valores comunes entre ellas y en que posicion estan.
7) Dadas tres colas, determinar en que orden fueron atendidos los clientes de cada una.
Ingrese el ejercicio al que ingresar ('0' PARA SALIR): 7

Para este ejercicio, dada tres colas y 'q' cantidad de minutos, informar el orden en que fueron atendidos los clientes

Primer cola:
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: *
Entrada invalida. Intente de nuevo.
Ingrese un entero o '/' para terminar: /
Segunda cola:
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: /
Tercer cola:
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: /
Ingresar tiempo(>0): 0



```
04-trabajoPractico-Colas > C main.c > main()
6 int main() {
8     while (1) {
19         switch (ejercicio) {
20             case 7:
39                 colasEjercicio7();
40                 break;
42             default:
43                 printf("Ese ejercicio no existe, solo del 2 al 2.\n\n");
44         }
}
```

Para este ejercicio, dada tres colas y 'q' cantidad de minutos, informar el orden en que fueron atendidos los clientes

Primer cola:
Ingrese un entero o '/' para terminar: 3
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: /
Segunda cola:
Ingrese un entero o '/' para terminar: 5
Ingrese un entero o '/' para terminar: 3
Ingrese un entero o '/' para terminar: /
Tercer cola:
Ingrese un entero o '/' para terminar: 10
Ingrese un entero o '/' para terminar: /
Ingresar tiempo(>0): 5

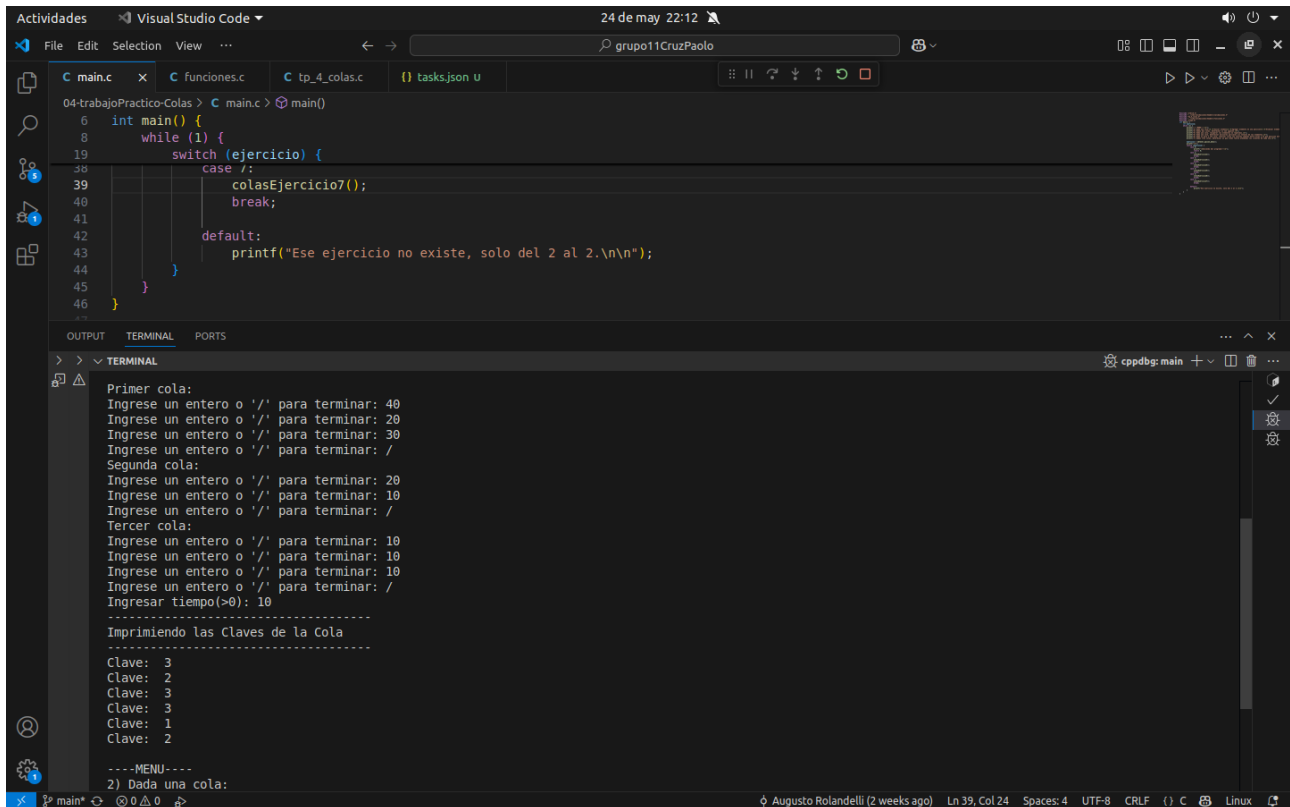
Imprimiendo las Claves de la Cola

Clave: 1
Clave: 2
Clave: 3

----MENU----

2) Dada una cola:
-buscar elemento
-Agregar elemento en una posicion
-Eliminar elemento todas las veces que aparezca
-Contar elementos de la cola
-Duplicar cola

GRUPO 11 – Correcciones Trabajo Práctico: COLAS



The image shows a Visual Studio Code editor window with a C program for a queue simulation. The program is named `main.c` and is located in the `04-trabajoPractico-Colas` directory. The code defines a `main` function that uses a `while` loop to process a series of commands. A `switch` statement handles different exercises, with `colasEjercicio7()` being called for exercise 7. A default case prints an error message. The terminal output shows the program's execution, including prompts for commands and the resulting queue state.

```
04-trabajoPractico-Colas > C main.c > main()
6  int main() {
8      while (1) {
19         switch (ejercicio) {
38             case 7:
39                 colasEjercicio7();
40                 break;
41             default:
42                 printf("Ese ejercicio no existe, solo del 2 al 2.\n\n");
43             }
44         }
45     }
46 }
```

OUTPUT TERMINAL PORTS

```
> > > TERMINAL
Primer cola:
Ingrese un entero o '/' para terminar: 40
Ingrese un entero o '/' para terminar: 20
Ingrese un entero o '/' para terminar: 30
Ingrese un entero o '/' para terminar: /
Segunda cola:
Ingrese un entero o '/' para terminar: 20
Ingrese un entero o '/' para terminar: 10
Ingrese un entero o '/' para terminar: /
Tercer cola:
Ingrese un entero o '/' para terminar: 10
Ingrese un entero o '/' para terminar: 10
Ingrese un entero o '/' para terminar: 10
Ingrese un entero o '/' para terminar: /
Ingresar tiempo(>0): 10
.....
Imprimiendo las Claves de la cola
.....
Clave: 3
Clave: 2
Clave: 3
Clave: 3
Clave: 1
Clave: 2

----MENU----
2) Dada una cola:
```

Augusto Rolandelli (2 weeks ago) Ln 39, Col 24 Spaces: 4 UTF-8 CRLF {} C Linux