



**Universidad Tres Culturas**



**PLANTEL “LONDRES”**

**Ingeniería en Sistemas Computacionales**

**Reporte de Prácticas**  
**Del 29 de julio al 2 de agosto.**

**Presenta:**  
*García Salas Natalia*

**Grupo: “A”**

**Turno: Matutino**

**Docente: José Guadalupe Sánchez  
Hernández**

**Asignatura: Estructura de Datos**

**Fecha de entrega: 6 de agosto de 2024.**

<b>Introducción</b>	<b>1</b>
<b>Desarrollo</b>	<b>1</b>
1. Pilas	1
Diagrama de Flujo	1
Código	3
Salida de escritorio	6
Documentación por bloques de código	7
<b>2. Colas</b>	<b>8</b>
Diagrama de flujo	8
Código	8
Salida de Escritorio	11
Documentación por bloques	12

## Introducción

Como estudiante de Ingeniería en Sistemas Computacionales es importante poner en práctica los conocimientos teóricos aprendidos en el aula de clases. Es por eso que se realizarán los siguientes ejercicios, para reforzar los conocimientos adquiridos en niveles anteriores de la carrera.

## Desarrollo

### 1. Pilas

#### Diagrama de Flujo

Inicio

|

v

Inicializar pila.cima = -1

|

v

system("cls")

|

v

eleccion = intOpcion()

|

v

+-----+

| eleccion != 4 |

| /\ |

| | |

```

|   v   |
| (switch eleccion) |
|   |   |
|   v   |
| eleccion == 1? -----+-----> LlenarPila(&pila)
|   |   |
|   v   |
| eleccion == 2? -----+-----> MostrarPila(&pila)
|   |   |
|   v   |
| eleccion == 3? -----+-----> EliminarPila(&pila)
|   |   |
|   v   |
| else -----+-----> printf("Opción incorrecta\n")
+-----+
|
v
FIN

```

LlenarPila(p):

```

|
v
(p->cima < 100 - 1) ?
|
+-----+
|           |
v           v
TRUE        FALSE
|           |
v           v
printf("Ingresa el dato:\n");
scanf("%d", &dato);
p->cima++;
p->dato[p->cima] = dato;
|
v
RETURN

```

MostrarPila(p):

```

|
v
(p->cima == -1) ?
|
+-----+
|           |
v           v
TRUE        FALSE

```

```

|
v
printf("La pila está vacía.\n");
printf("Contenido de la pila:\n");
for (i = p->cima; i >= 0; i--) {
    printf("%d\n", p->dato[i]);
}

|
v
RETURN

```

EliminarPila(p):

```

|
v
(p->cima == -1) ?
|
+-----+
|         |
v         v
TRUE      FALSE
|         |
v         v
printf("La pila está vacía.\n");
printf("Último elemento eliminado.\n");

|
v
RETURN

```

intOpcion():

```

|
v
printf("\nSeleccione una opción:\n");
printf("1. Nuevo nodo\n");
printf("2. Mostrar pila\n");
printf("3. Eliminar pila\n");
printf("4. Salir\n");
scanf("%d", &opcion);
|
v
RETURN opcion

```

## Código

//Programa hecho por @Natalia Garcia  
//24/07/24

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct Pila {
    int dato[100];
    int cima;
};
```

```
// Prototipos de funciones
int intOpcion(void);
void LlenarPila(struct Pila *p);
void MostrarPila(struct Pila *p);
void EliminarPila(struct Pila *p);
```

```
int main() {
    struct Pila pila;
    int eleccion;
    pila.cima = -1;
    system("cls");
    do{
        switch(eleccion){
            case 1: //Agregar elementos
                LlenarPila(&pila);
                break;
            case 2: //Mostrar pila
                MostrarPila(&pila);
                break;
            case 3: //Eliminar último elemento
                EliminarPila(&pila);
                break;
            default:
                if (eleccion < 1 || eleccion > 6) {
                    printf("Opción incorrecta\n");
                }
                break;
        }
    }while((eleccion = intOpcion()) != 4);
    return 0;
}
```

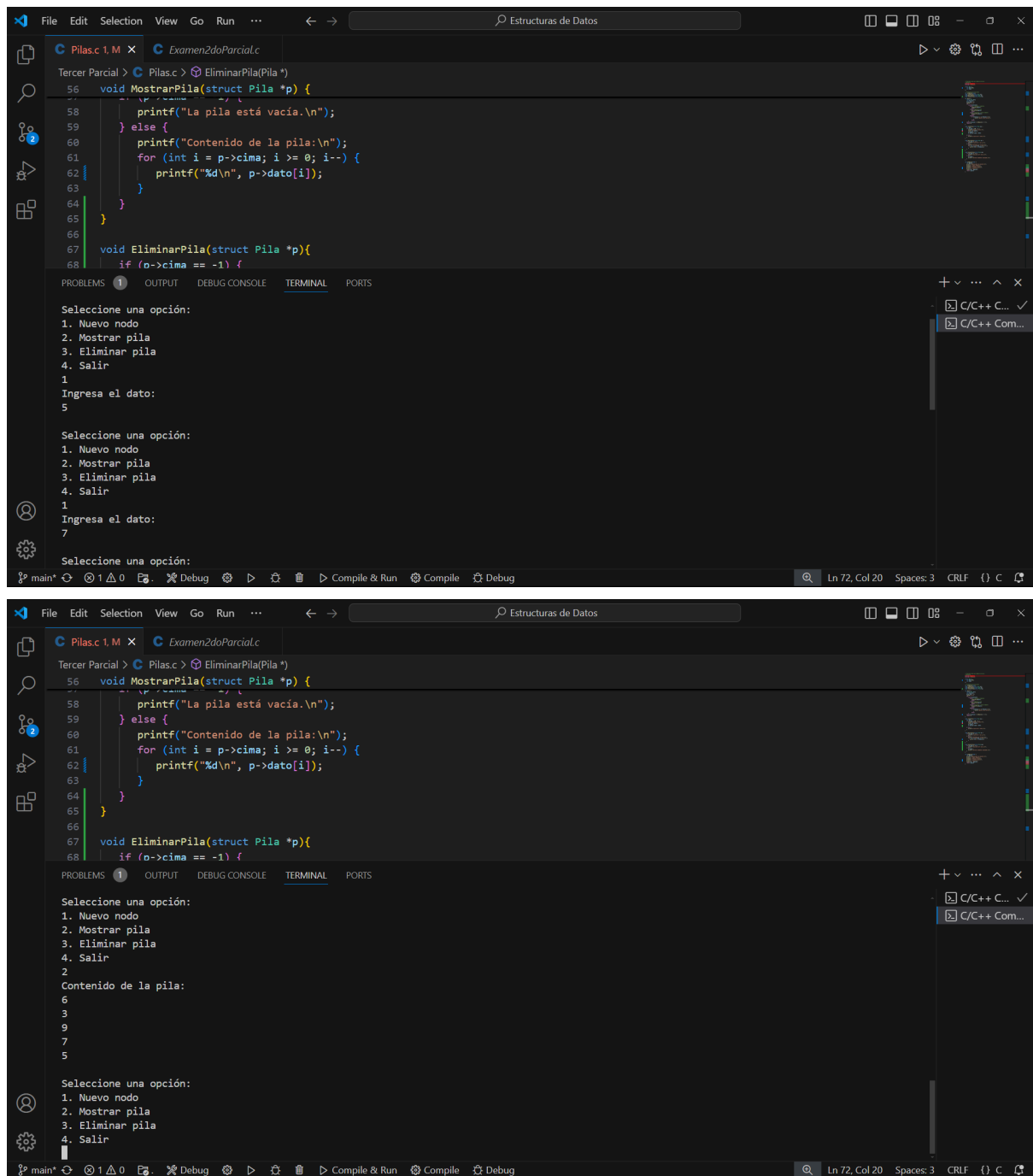
```
void LlenarPila(struct Pila *p) {
    int dato;
    if (p->cima < 100 - 1) {
        printf("Ingresa el dato:\n");
        scanf("%d",&dato);
        p->cima++;
        p->dato[p->cima] = dato;
    }
```

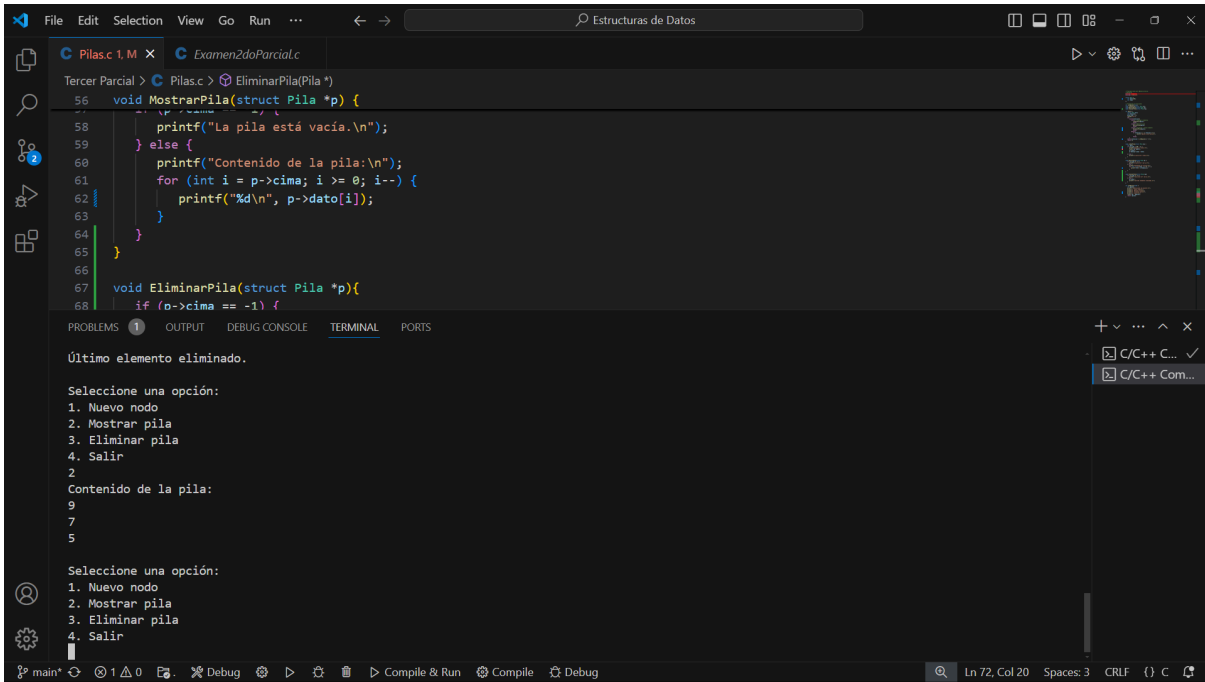
```
} else {
    printf("La pila está llena.\n");
}
}

void MostrarPila(struct Pila *p) {
    if (p->cima == -1) {
        printf("La pila está vacía.\n");
    } else {
        printf("Contenido de la pila:\n");
        for (int i = p->cima; i >= 0; i--) {
            printf("%d\n", p->dato[i]);
        }
    }
}

void EliminarPila(struct Pila *p){
    if (p->cima == -1) {
        printf("\nLa pila está vacía.\n");
    } else {
        p->cima--;
        printf("\nÚltimo elemento eliminado.\n");
    }
}

int intOpcion(void) {
    int opcion;
    printf("\nSeleccione una opción:\n");
    printf("1. Nuevo nodo\n");
    printf("2. Mostrar pila\n");
    printf("3. Eliminar pila\n");
    printf("4. Salir\n");
    scanf("%d", &opcion);
    return opcion;
}
```





```
File Edit Selection View Go Run ... ← → Estructuras de Datos
C Pilas.c 1, M x Examen2doParcial.c
Tercer Parcial > C Pilas.c > EliminarPila(Pila *)
56 void MostrarPila(struct Pila *p) {
57     if (p->cima == -1) {
58         printf("La pila está vacía.\n");
59     } else {
60         printf("Contenido de la pila:\n");
61         for (int i = p->cima; i >= 0; i--) {
62             printf("%d\n", p->dato[i]);
63         }
64     }
65 }
66
67 void EliminarPila(struct Pila *p){
68     if (p->cima == -1) {
69         printf("La pila está vacía.\n");
70     } else {
71         p->cima--;
72     }
73 }
74
75 int main() {
76     Pila p;
77     inicializarPila(&p);
78     while (1) {
79         printf("Seleccione una opción:\n");
80         printf("1. Nuevo nodo\n");
81         printf("2. Mostrar pila\n");
82         printf("3. Eliminar pila\n");
83         printf("4. Salir\n");
84         int opcion;
85         scanf("%d", &opcion);
86         switch (opcion) {
87             case 1:
88                 NuevoNodo(&p);
89                 break;
90             case 2:
91                 MostrarPila(&p);
92                 break;
93             case 3:
94                 EliminarPila(&p);
95                 break;
96             case 4:
97                 return 0;
98             default:
99                 printf("Opción no válida.\n");
100            }
101        }
102    }
103    return 0;
104 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Último elemento eliminado.

Seleccione una opción:

1. Nuevo nodo  
2. Mostrar pila  
3. Eliminar pila  
4. Salir

2

Contenido de la pila:

9  
7  
5

Seleccione una opción:

1. Nuevo nodo  
2. Mostrar pila  
3. Eliminar pila  
4. Salir

3

## Documentación por bloques de código

- Bloque Verde

Declaración de variables

- Bloque Rojo

Lectura de datos

- Bloque Azul

Operaciones

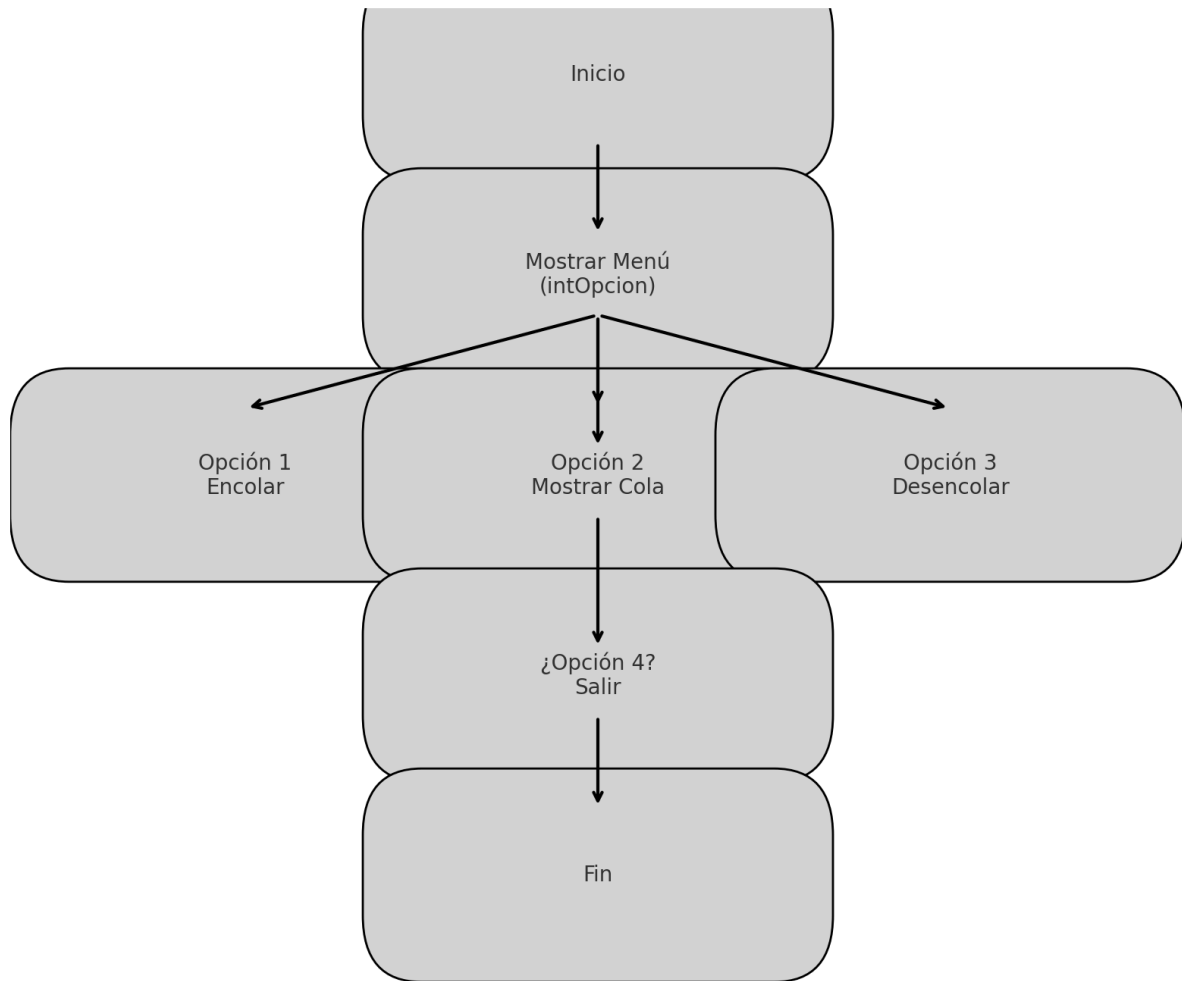
- Bloque Naranja

Resultado final



## 2. Colas

### Diagrama de flujo



### Código

```
//Programa hecho por @Natalia Garcia
//31/07/24
#include <stdio.h>
#include <stdlib.h>
```

```
struct Cola {
    int dato;
    struct Cola *ptrSiguiente;
};

// Prototipos de funciones
int intOpcion(void);
void Ingresar(struct Cola **frente, struct Cola **final);
void MostrarCola(struct Cola *frente);
void Eliminar(struct Cola **frente, struct Cola **final);

int main() {
    struct Cola *frente = NULL;
    struct Cola *final = NULL;
    int opc;
    do{
        switch(opc){
            case 1: //Agregar elementos
                Ingresar(&frente, &final);
                break;
            case 2: //Mostrar cola
                MostrarCola(frente);
                break;
            case 3: //Eliminar primer elemento
                Eliminar(&frente, &final);
                break;
            default:
                if (opc < 1 || opc > 4) {
                    printf("Opción incorrecta\n");
                }
                break;
        }
    }while((opc = intOpcion()) != 4);
    return 0;
}

void Ingresar(struct Cola **frente, struct Cola **final) {
    int dato;
    struct Cola *nuevoNodo = (struct Cola*) malloc(sizeof(struct Cola));
    if(nuevoNodo == NULL) {
        printf("No hay memoria disponible.\n");
        return;
    }
    printf("Ingrese el dato: ");
    scanf("%d",&dato);
    nuevoNodo->dato = dato;
    nuevoNodo->ptrSiguiente = NULL;
```

```
if (*final == NULL) {
    *frente = nuevoNodo;
} else {
    (*final)->ptrSiguiente = nuevoNodo;
}
*final = nuevoNodo;
}

void MostrarCola(struct Cola *frente) {
    if (frente == NULL) {
        printf("La cola está vacía.\n");
    } else {
        struct Cola *actual = frente;
        printf("Contenido de la cola:\n");
        while(actual != NULL){
            printf("%d\n", actual->dato);
            actual = actual->ptrSiguiente;
        }
    }
}

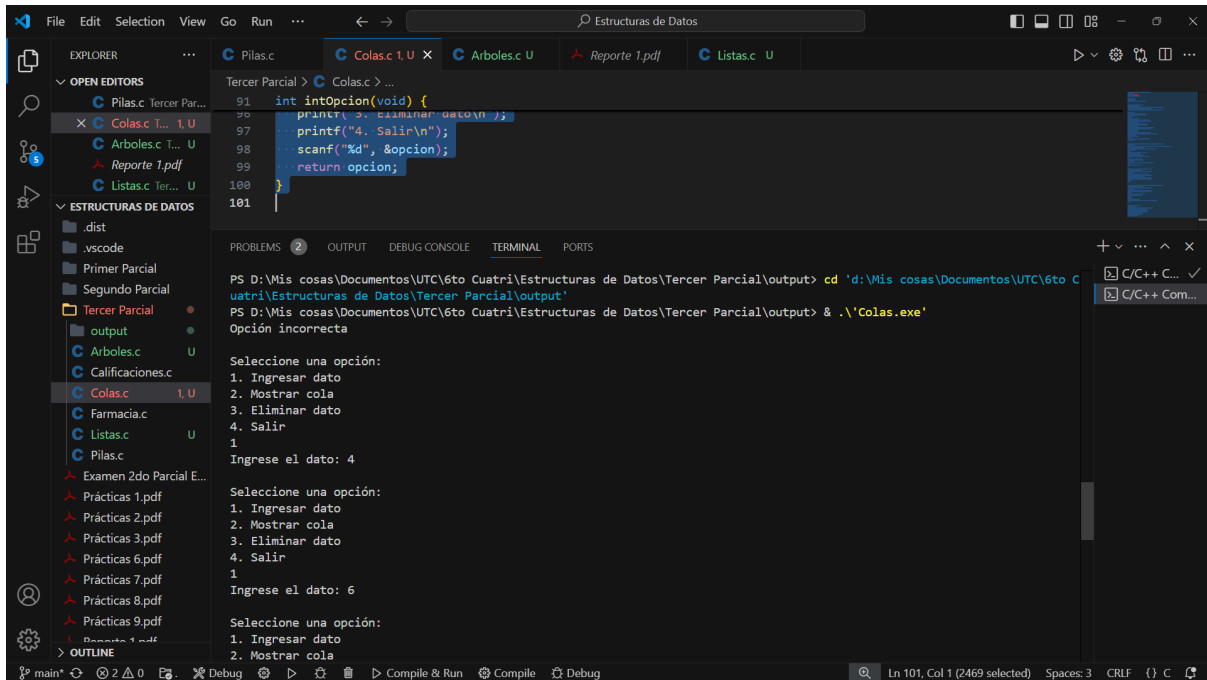
void Eliminar(struct Cola **frente, struct Cola **final){
    if (*frente == NULL) {
        printf("\nLa cola está vacía.\n");
    } else {
        struct Cola *ptrTemp = *frente;
        *frente = (*frente)->ptrSiguiente;

        if (*frente == NULL) {
            *final = NULL;
        }

        free(ptrTemp);
        printf("\nPrimer elemento eliminado.\n");
    }
}

int intOpcion(void) {
    int opcion;
    printf("\nSeleccione una opción:\n");
    printf("1. Ingresar dato\n");
    printf("2. Mostrar cola\n");
    printf("3. Eliminar dato\n");
    printf("4. Salir\n");
    scanf("%d", &opcion);
    return opcion;
}
```

## Salida de Escritorio



File Edit Selection View Go Run ... Estructuras de Datos

EXPLORER

- OPEN EDITORS
  - Pilas.c Tercer Par...
  - Colas.c T... 1.U
  - Arboles.c T... U
  - Reporte 1.pdf
  - Listas.c Ter... U
- ESTRUCTURAS DE DATOS
  - .dist
  - .vscode
  - Primer Parcial
  - Segundo Parcial
  - Tercer Parcial
    - output
    - Arboles.c U
    - Calificaciones.c
    - Colas.c 1.U
    - Farmacia.c
    - Listas.c U
    - Pilas.c
  - Examen 2do Parcial E...
  - Prácticas 1.pdf
  - Prácticas 2.pdf
  - Prácticas 3.pdf
  - Prácticas 6.pdf
  - Prácticas 7.pdf
  - Prácticas 8.pdf
  - Prácticas 9.pdf
- OUTLINE

Tercer Parcial > Colas.c > ...

```

91 int intOpcion(void) {
92     printf("3. Eliminar dato\n");
93     printf("4. Salir\n");
94     scanf("%d", &opcion);
95     return opcion;
96 }
97
98
99
100
101

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Mis cosas\Documentos\UTC\6to Cuatri\Estructuras de Datos\Tercer Parcial\output> cd 'd:\Mis cosas\Documentos\UTC\6to Cuatri\Estructuras de Datos\Tercer Parcial\output' & .\Colas.exe

Opción incorrecta

Seleccione una opción:

1. Ingresar dato
2. Mostrar cola
3. Eliminar dato
4. Salir

1

Ingrese el dato: 4

Seleccione una opción:

1. Ingresar dato
2. Mostrar cola
3. Eliminar dato
4. Salir

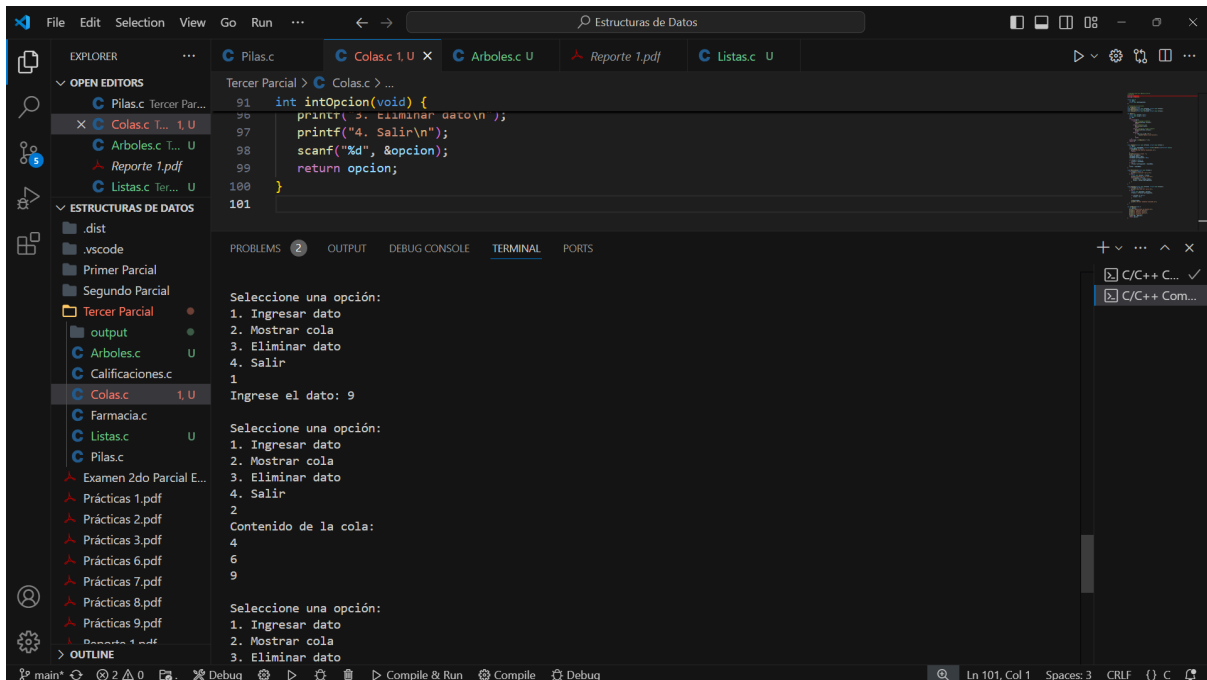
1

Ingrese el dato: 6

Seleccione una opción:

1. Ingresar dato
2. Mostrar cola

Ln 101, Col 1 (2469 selected) Spaces: 3 CRLF {} C



File Edit Selection View Go Run ... Estructuras de Datos

EXPLORER

- OPEN EDITORS
  - Pilas.c Tercer Par...
  - Colas.c T... 1.U
  - Arboles.c T... U
  - Reporte 1.pdf
  - Listas.c Ter... U
- ESTRUCTURAS DE DATOS
  - .dist
  - .vscode
  - Primer Parcial
  - Segundo Parcial
  - Tercer Parcial
    - output
    - Arboles.c U
    - Calificaciones.c
    - Colas.c 1.U
    - Farmacia.c
    - Listas.c U
    - Pilas.c
  - Examen 2do Parcial E...
  - Prácticas 1.pdf
  - Prácticas 2.pdf
  - Prácticas 3.pdf
  - Prácticas 6.pdf
  - Prácticas 7.pdf
  - Prácticas 8.pdf
  - Prácticas 9.pdf
- OUTLINE

Tercer Parcial > Colas.c > ...

```

91 int intOpcion(void) {
92     printf("3. Eliminar dato\n");
93     printf("4. Salir\n");
94     scanf("%d", &opcion);
95     return opcion;
96 }
97
98
99
100
101

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Seleccione una opción:

1. Ingresar dato
2. Mostrar cola
3. Eliminar dato
4. Salir

1

Ingrese el dato: 9

Seleccione una opción:

1. Ingresar dato
2. Mostrar cola
3. Eliminar dato
4. Salir

2

Contenido de la cola:

4

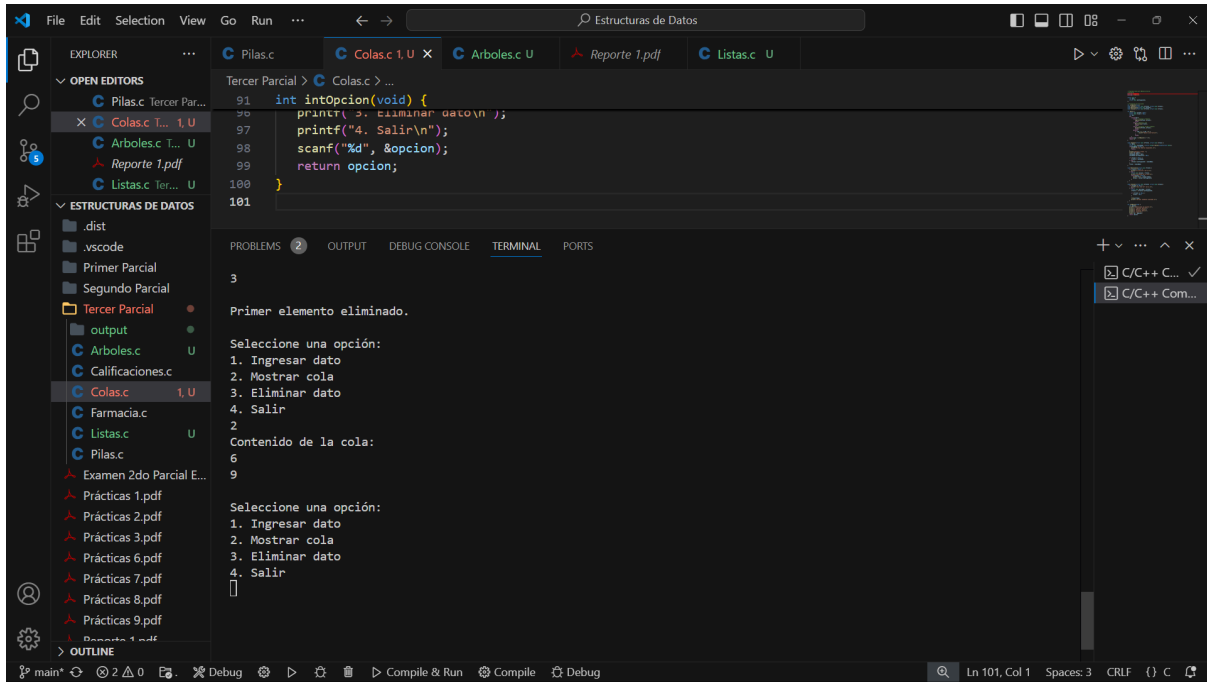
6

9

Seleccione una opción:

1. Ingresar dato
2. Mostrar cola
3. Eliminar dato

Ln 101, Col 1 Spaces: 3 CRLF {} C



```

91 int intOpcion(void) {
92     printf("3. Eliminar dato\n");
93     printf("4. Salir\n");
94     scanf("%d", &opcion);
95     return opcion;
96 }
97
98
99
100
101

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

3

Primer elemento eliminado.

Seleccione una opción:

1. Ingresar dato

2. Mostrar cola

3. Eliminar dato

4. Salir

2

Contenido de la cola:

6

9

Seleccione una opción:

1. Ingresar dato

2. Mostrar cola

3. Eliminar dato

4. Salir

## Documentación por bloques

### ● Bloque Verde

Declaración de variables

### ● Bloque Rojo

Lectura de datos

### ● Bloque Azul

Operaciones

### ● Bloque Naranja

Resultado final