

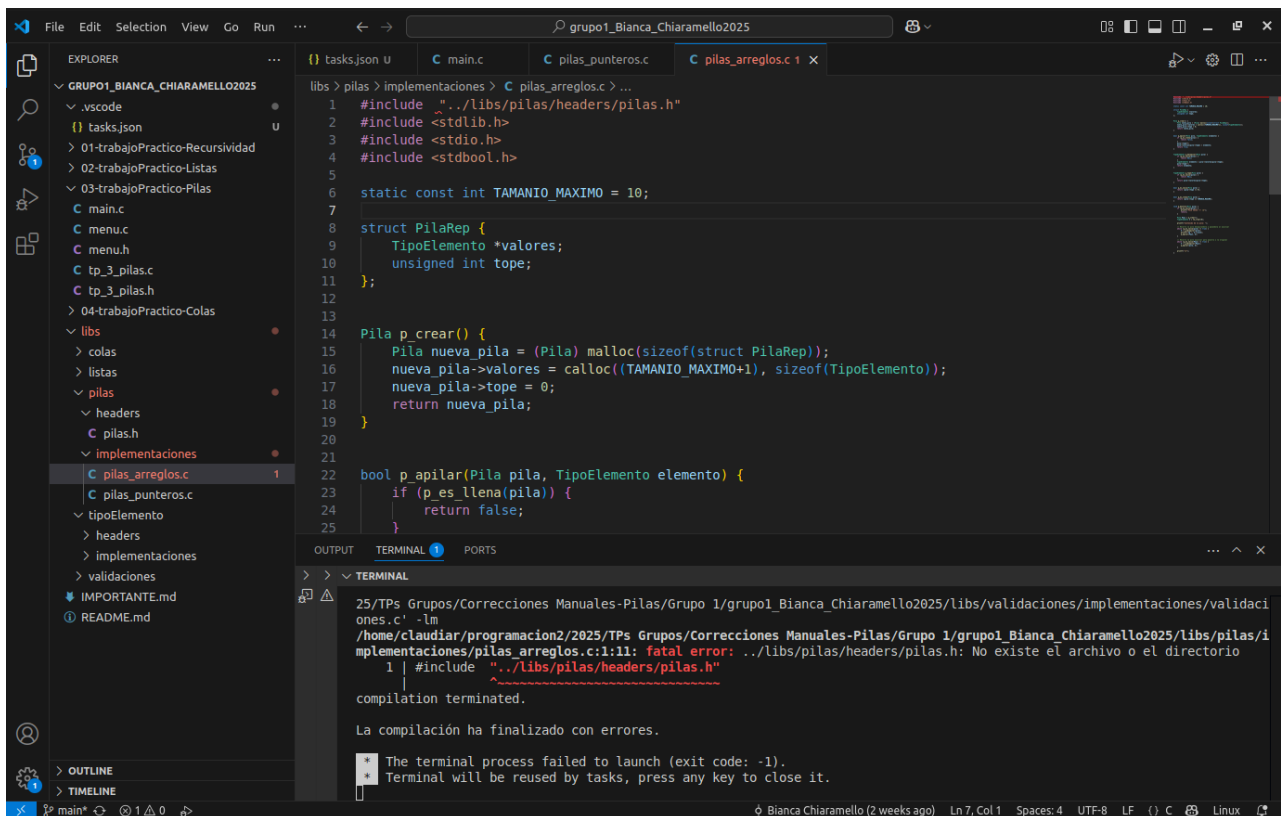
RESULTADO DE LA CORRECCIÓN: **APROBADO** —

## OBSERVACIONES GENERALES A TODOS LOS GRUPOS

En caso de no haber resultado para mostrar, por ejemplo no se encontró coincidencias en lo buscado, indicarlo por pantalla. Si no se muestra nada no queda claro el resultado.

## OBSERVACIONES

Include mal realizado. No permite pila vacía.



```
File Edit Selection View Go Run ... grupo1_Bianca_Chiamello2025
EXPLORER
  GRUPO1_BIANCA_CHIARAMELLO2025
    .vscode
    tasks.json
    01-trabajoPractico-Recursividad
    02-trabajoPractico-Listas
    03-trabajoPractico-Pilas
      main.c
      menu.c
      menu.h
      tp_3_pilas.c
      tp_3_pilas.h
    04-trabajoPractico-Colas
      colas
      listas
      pilas
        headers
        pilas.h
        implementaciones
          pilas_arreglos.c
          pilas_punteros.c
        tipoElemento
        headers
        implementaciones
        validaciones
      IMPORTANTE.md
      README.md
  libs
    colas
    listas
    pilas
      headers
      pilas.h
      implementaciones
      pilas_arreglos.c
      pilas_punteros.c
      tipoElemento
      headers
      implementaciones
      validaciones
  IMPORTANTE.md
  README.md
  OUTLINE
  TIMELINE
  main* 0

tasks.json
main.c
pilas_punteros.c
pilas_arreglos.c 1

libs > pilas > implementaciones > C pilas_arreglos.c > ...
1 #include "../libs/pilas/headers/pilas.h"
2 #include <stdlib.h>
3 #include <stdio.h>
4 #include <stdbool.h>
5
6 static const int TAMANIO_MAXIMO = 10;
7
8 struct PilaRep {
9     TipoElemento *valores;
10    unsigned int tope;
11 };
12
13
14 Pila p_crear() {
15     Pila nueva_pila = (Pila) malloc(sizeof(struct PilaRep));
16     nueva_pila->valores = calloc((TAMANIO_MAXIMO+1), sizeof(TipoElemento));
17     nueva_pila->tope = 0;
18     return nueva_pila;
19 }
20
21
22 bool p_apilar(Pila pila, TipoElemento elemento) {
23     if (p_es_llena(pila)) {
24         return false;
25     }
26 }

OUTPUT TERMINAL PORTS
> > TERMINAL
25/TPs Grupos/Correcciones Manuales-Pilas/Grupo 1/grupo1_Bianca_Chiamello2025/libs/validaciones/implementaciones/validaciones.c' -lm
/home/clauiar/programacion2/2025/TPs Grupos/Correcciones Manuales-Pilas/Grupo 1/grupo1_Bianca_Chiamello2025/libs/pilas/implementaciones/pilas_arreglos.c:1:11: fatal error: ../libs/pilas/headers/pilas.h: No existe el archivo o el directorio
1 | #include "../libs/pilas/headers/pilas.h"
  | ~~~~~^~~~~~
compilation terminated.

La compilación ha finalizado con errores.

* The terminal process failed to launch (exit code: -1).
* Terminal will be reused by tasks, press any key to close it.
```

## GRUPO 1 – Correcciones Trabajo Práctico: PILAS

```
void main() {
    do {
        switch (opcion) {
            break;
            case 5:
                menuPunto5();
                break;
            case 6:
                menuPunto6();
                break;
            case 7:
                menuPunto7();
                break;
            case 8:
                menuPunto8();
                break;
            case 0:
                printf("Saliendo...\n");
                system("pause");
                break;
            default:

```

=== MENU PRINCIPAL ===  
2. Ejercicio 2  
3. Ejercicio 3  
4. Ejercicio 4  
5. Ejercicio 5  
6. Ejercicio 6  
7. Ejercicio 7  
8. Ejercicio 8  
0. Salir  
> Seleccione una opcion: 6  
sh: 1: cls: not found  
Ingrese la cantidad de elementos a cargar en la pila (max 10)  
> 0  
Ingrese un valor adecuado!  
->

Ejercicio 6: no se indica de manera clara cuál es la pila original.

```
#include "menu.h"

void main() {
    int opcion;

    do {
        printf("\n=== MENU PRINCIPAL ===\n");
        printf("2. Ejercicio 2 \n");
        printf("3. Ejercicio 3 \n");
        printf("4. Ejercicio 4 \n");
        printf("5. Ejercicio 5 \n");
        printf("6. Ejercicio 6 \n");
        printf("7. Ejercicio 7 \n");
        printf("8. Ejercicio 8 \n");
        printf("0. Salir\n");
        printf("> Seleccione una opcion: ");
        opcion = ingresoDatosNumericosPositivos("\nEntrada erronea.\n-> ");
        system("cls");

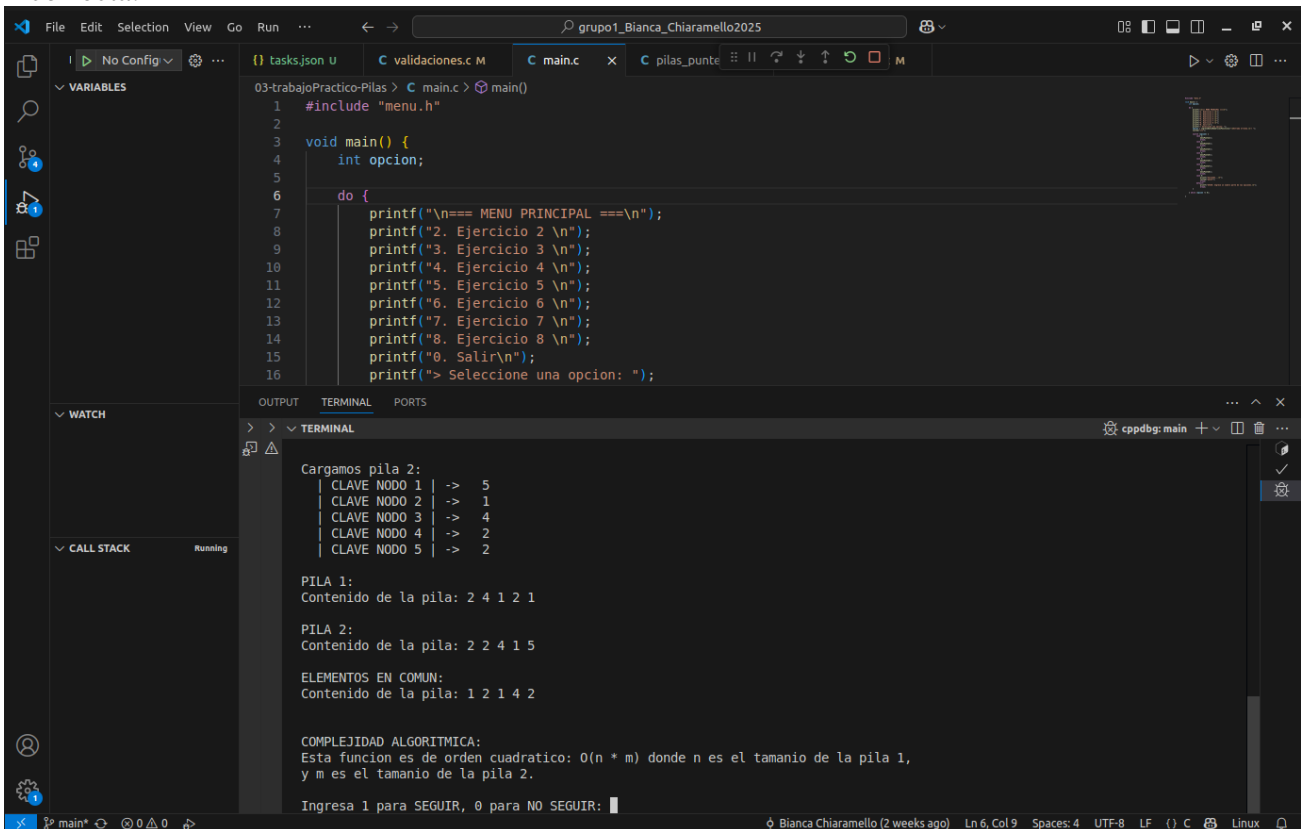
        switch (opcion) {
            case 2:

```

PILA:  
Contenido de la pila: 3 2 1  
  
RESOLUCION RECURSIVA -- 1  
RESOLUCION ITERATIVA -- 2  
-> 2  
  
CLAVE A ELIMINAR: 2  
Contenido de la pila: 3 1  
Contenido de la pila: 3 2 1  
  
COMPLEJIDAD ALGORITMICA:  
Esta funcion es de orden lineal: O(n), donde n es la cantidad de elementos en la pila original.  
Ingresar 1 para SEGUIR, 0 para NO SEGUIR: 1

## GRUPO 1 – Correcciones Trabajo Práctico: PILAS

Ejercicio 7: no permite pilas de diferente tamaño. Si alguna pila tiene elementos repetidos que son comunes a la otra pila, los repite en la pila resultado, no los detecta como iguales. Complejidad incorrecta.



```
03-trabajoPractico-Pilas > C main.c > main()
1 #include "menu.h"
2
3 void main() {
4     int opcion;
5
6     do {
7         printf("\n== MENU PRINCIPAL ==\n");
8         printf("2. Ejercicio 2 \n");
9         printf("3. Ejercicio 3 \n");
10        printf("4. Ejercicio 4 \n");
11        printf("5. Ejercicio 5 \n");
12        printf("6. Ejercicio 6 \n");
13        printf("7. Ejercicio 7 \n");
14        printf("8. Ejercicio 8 \n");
15        printf("0. Salir\n");
16        printf("> Seleccione una opcion: ");
```

OUTPUT TERMINAL

```
Cargamos pila 2:
| CLAVE NODO 1 | -> 5
| CLAVE NODO 2 | -> 1
| CLAVE NODO 3 | -> 4
| CLAVE NODO 4 | -> 2
| CLAVE NODO 5 | -> 2

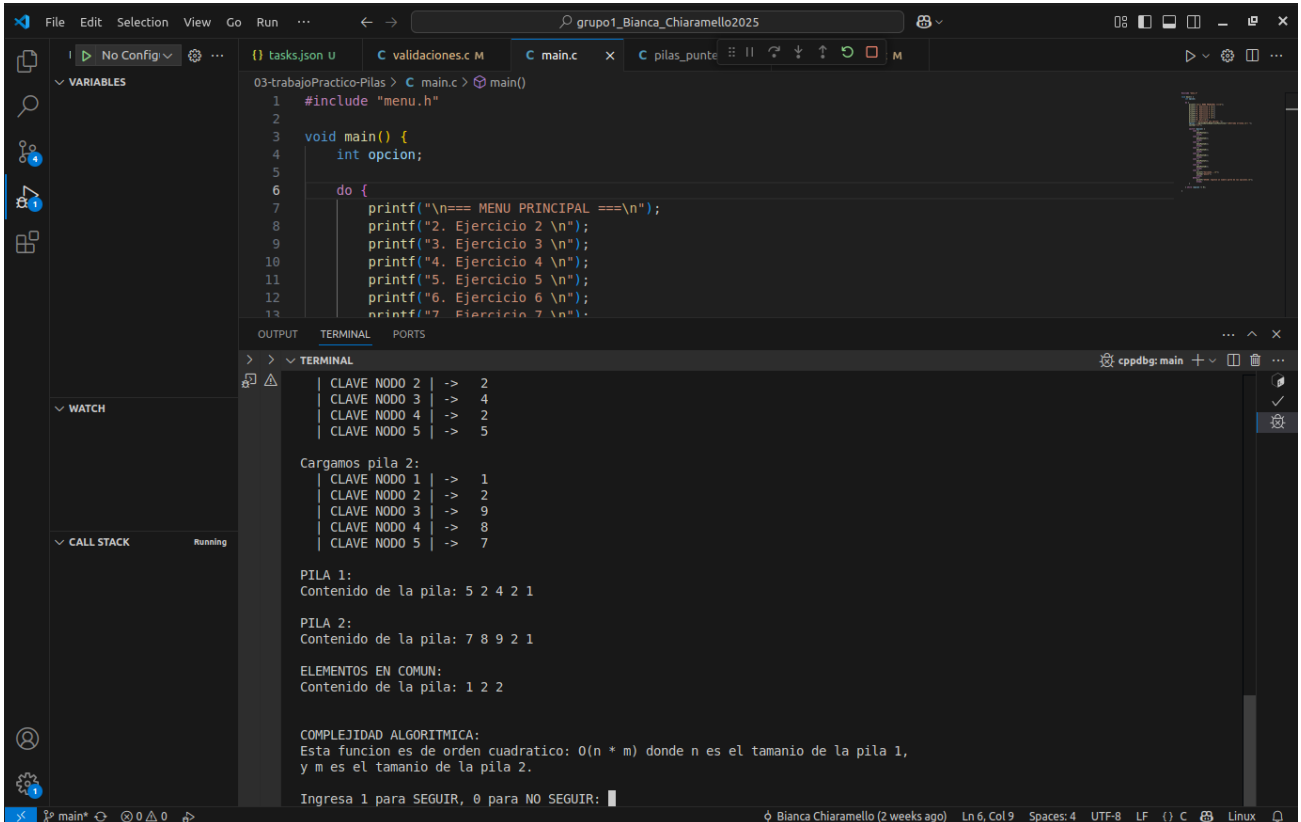
PILA 1:
Contenido de la pila: 2 4 1 2 1

PILA 2:
Contenido de la pila: 2 2 4 1 5

ELEMENTOS EN COMUN:
Contenido de la pila: 1 2 1 4 2

COMPLEJIDAD ALGORITMICA:
Esta funcion es de orden cuadratico: O(n * m) donde n es el tamaño de la pila 1,
y m es el tamaño de la pila 2.

Ingresa 1 para SEGUIR, 0 para NO SEGUIR:
```



```
03-trabajoPractico-Pilas > C main.c > main()
1 #include "menu.h"
2
3 void main() {
4     int opcion;
5
6     do {
7         printf("\n== MENU PRINCIPAL ==\n");
8         printf("2. Ejercicio 2 \n");
9         printf("3. Ejercicio 3 \n");
10        printf("4. Ejercicio 4 \n");
11        printf("5. Ejercicio 5 \n");
12        printf("6. Ejercicio 6 \n");
13        printf("7. Ejercicio 7 \n");
```

OUTPUT TERMINAL

```
CLAVE NODO 2 | -> 2
CLAVE NODO 3 | -> 4
CLAVE NODO 4 | -> 2
CLAVE NODO 5 | -> 5

Cargamos pila 2:
| CLAVE NODO 1 | -> 1
| CLAVE NODO 2 | -> 2
| CLAVE NODO 3 | -> 9
| CLAVE NODO 4 | -> 8
| CLAVE NODO 5 | -> 7

PILA 1:
Contenido de la pila: 5 2 4 2 1

PILA 2:
Contenido de la pila: 7 8 9 2 1

ELEMENTOS EN COMUN:
Contenido de la pila: 1 2 2

COMPLEJIDAD ALGORITMICA:
Esta funcion es de orden cuadratico: O(n * m) donde n es el tamaño de la pila 1,
y m es el tamaño de la pila 2.

Ingresa 1 para SEGUIR, 0 para NO SEGUIR:
```