

**RESULTADO DE LA CORRECCIÓN: DESAPROBADO**

## OBSERVACIONES

Modularizar el menú, es muy complicado para corregir el código. Pide confirmación de carga de cada elemento y acumula los valores en la pila cuando se ingresa nuevamente a la opción de carga, como ya se mencionó en el TP de Listas. Carga el último valor aunque se indica que no lo haga. En el ejercicio 6 aunque muestra que no pierde la pila original durante la primera eliminación, cuando se ingresa a eliminar elemento desde otra opción del menú, muestra la pila luego de haber eliminado el elemento anterior porque pisan la pila original. No responden sobre complejidad. En el ejercicio 7 no me permite cargar la Pila 2 porque sale del ciclo y genera un error. Igual comportamiento en el ejercicio 8

The screenshot shows a C++ IDE with a dark theme. The editor window displays a file named `main.c` with the following code:

```
1 #include "pilas.h"
2 #include "funciones.h"
3 #include "tipo_elemento.h"
4 #include "validaciones.h"
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <stdbool.h>
9 #include <time.h>
10
```

The interface includes a sidebar on the left with icons for Explorer, Search, Source Control, and Run and Debug. The top menu bar shows File, Edit, Selection, View, Go, Run, Terminal, and Help. The bottom status bar indicates the current line and column (Ln1, Col1), character count (Spaces: 4), encoding (UTF-8), and the active file (main.c).

The Run and Debug window is open, showing the output of the program. The output is as follows:

```
MENU EJERCICIO 6 'TP PILAS':
1.  Agregar elementos a la PILA (MANUALMENTE).
2.  Agregar elementos a la PILA (ALEATORIAMENTE).
3.  Obtener la PILA sin X elemento (ITERATIVO).
4.  Obtener la PILA sin X elemento (RECURSIVO).
5.  Mostrar la PILA actual.
0.  VOLVER AL MENU DEL SISTEMA.

Ingrese el numero de la accion a realizar: 1

sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 1

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1

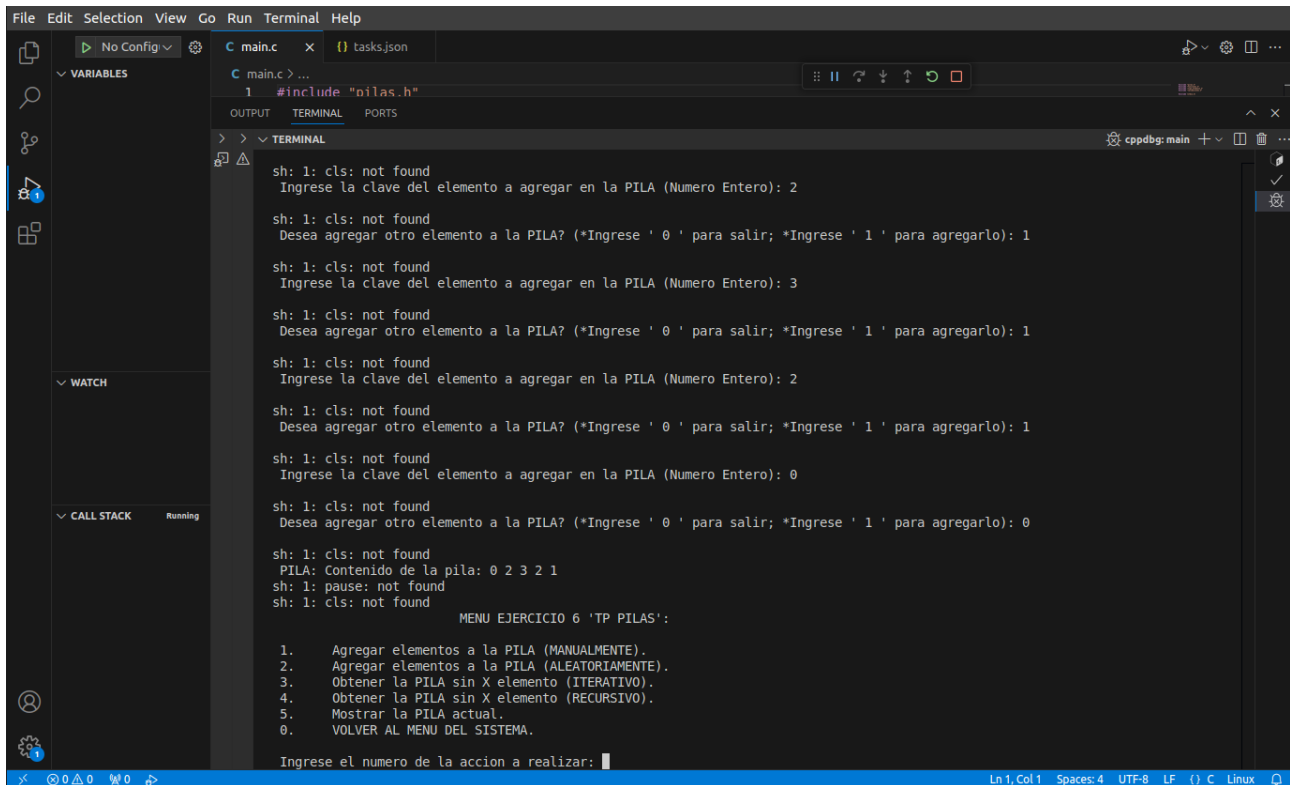
sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 2

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1

sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 3

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1
```

## GRUPO 2 – Correcciones Trabajo Práctico: PILAS



```
File Edit Selection View Go Run Terminal Help
C main.c x tasks.json
1 #include "pilas.h"
OUTPUT TERMINAL PORTS
sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 2

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1

sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 3

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1

sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 2

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1

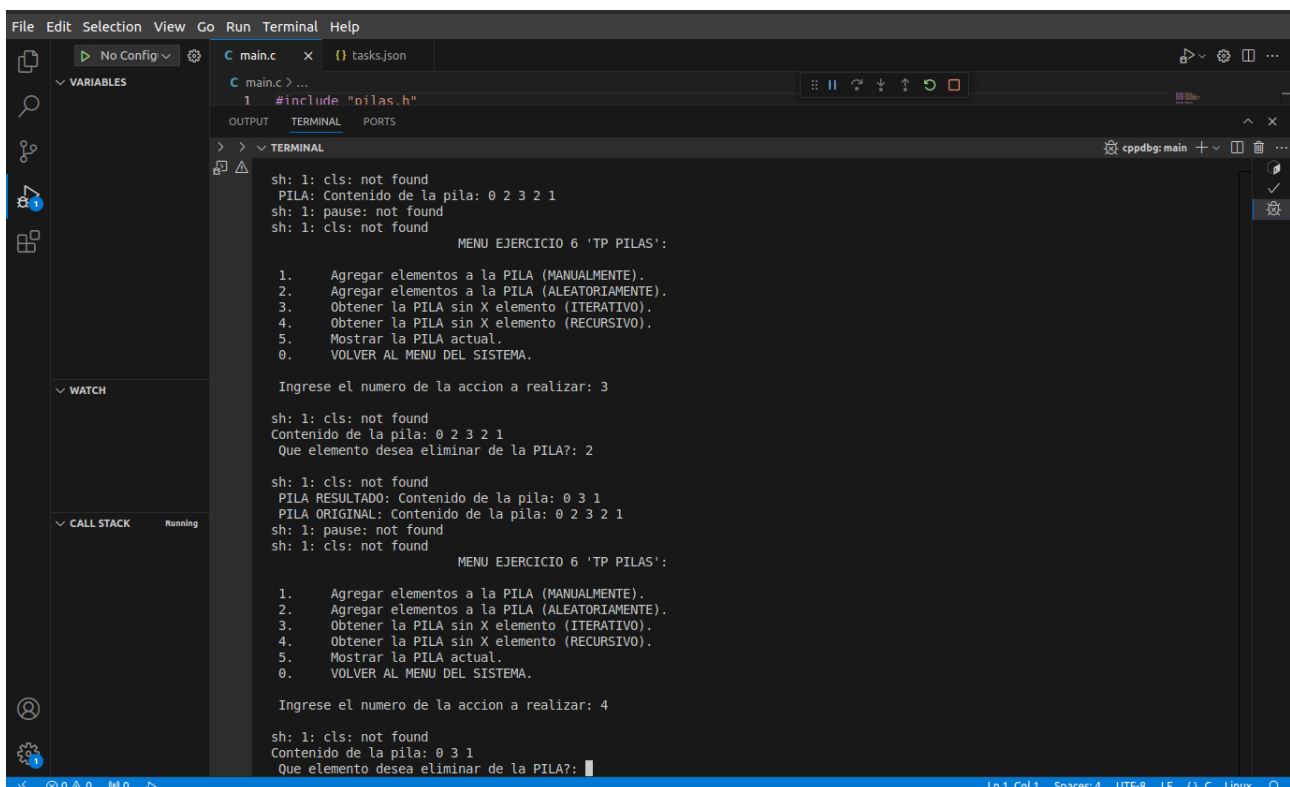
sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 0

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 0

sh: 1: cls: not found
PILA: Contenido de la pila: 0 2 3 2 1
sh: 1: pause: not found
sh: 1: cls: not found
MENU EJERCICIO 6 'TP PILAS':

1. Agregar elementos a la PILA (MANUALMENTE).
2. Agregar elementos a la PILA (ALEATORIAMENTE).
3. Obtener la PILA sin X elemento (ITERATIVO).
4. Obtener la PILA sin X elemento (RECURSIVO).
5. Mostrar la PILA actual.
0. VOLVER AL MENU DEL SISTEMA.

Ingrese el numero de la accion a realizar: 
```



```
File Edit Selection View Go Run Terminal Help
C main.c x tasks.json
1 #include "pilas.h"
OUTPUT TERMINAL PORTS
sh: 1: cls: not found
PILA: Contenido de la pila: 0 2 3 2 1
sh: 1: pause: not found
sh: 1: cls: not found
MENU EJERCICIO 6 'TP PILAS':

1. Agregar elementos a la PILA (MANUALMENTE).
2. Agregar elementos a la PILA (ALEATORIAMENTE).
3. Obtener la PILA sin X elemento (ITERATIVO).
4. Obtener la PILA sin X elemento (RECURSIVO).
5. Mostrar la PILA actual.
0. VOLVER AL MENU DEL SISTEMA.

Ingrese el numero de la accion a realizar: 3

sh: 1: cls: not found
Contenido de la pila: 0 2 3 2 1
Que elemento desea eliminar de la PILA?: 2

sh: 1: cls: not found
PILA RESULTADO: Contenido de la pila: 0 3 1
PILA ORIGINAL: Contenido de la pila: 0 2 3 2 1
sh: 1: pause: not found
sh: 1: cls: not found
MENU EJERCICIO 6 'TP PILAS':

1. Agregar elementos a la PILA (MANUALMENTE).
2. Agregar elementos a la PILA (ALEATORIAMENTE).
3. Obtener la PILA sin X elemento (ITERATIVO).
4. Obtener la PILA sin X elemento (RECURSIVO).
5. Mostrar la PILA actual.
0. VOLVER AL MENU DEL SISTEMA.

Ingrese el numero de la accion a realizar: 4

sh: 1: cls: not found
Contenido de la pila: 0 3 1
Que elemento desea eliminar de la PILA?: 
```

## GRUPO 2 – Correcciones Trabajo Práctico: PILAS

The image shows a C++ IDE with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The left sidebar contains icons for Explorer, Search, Source Control, and Run and Debug. The main editor area shows a C++ file named `main.c` with the following content:

```
C main.c > ...
1 #include "pilas.h"
```

The terminal window, titled `cpplib: main`, shows the output of the program. It starts with a menu titled `MENU EJERCICIO 6 'TP PILAS':` with options 1 to 5. The user enters 4, and the program outputs the stack content and asks for an element to remove. The user enters 3, and the program outputs the updated stack content.

```
PILA RESULTADO: Contenido de la pila: 0 3 1
PILA ORIGINAL: Contenido de la pila: 0 2 3 2 1
sh: 1: pause: not found
sh: 1: cls: not found

MENU EJERCICIO 6 'TP PILAS':

1.  Agregar elementos a la PILA (MANUALMENTE).
2.  Agregar elementos a la PILA (ALEATORIAMENTE).
3.  Obtener la PILA sin X elemento (ITERATIVO).
4.  Obtener la PILA sin X elemento (RECURSIVO).
5.  Mostrar la PILA actual.
0.  VOLVER AL MENU DEL SISTEMA.

Ingrese el numero de la accion a realizar: 4

sh: 1: cls: not found
Contenido de la pila: 0 3 1
Que elemento desea eliminar de la PILA?: 1

sh: 1: cls: not found
PILA RESULTADO: Contenido de la pila: 0 3
PILA ORIGINAL: Contenido de la pila: 0 3 1
sh: 1: pause: not found
sh: 1: cls: not found

MENU EJERCICIO 6 'TP PILAS':

1.  Agregar elementos a la PILA (MANUALMENTE).
2.  Agregar elementos a la PILA (ALEATORIAMENTE).
3.  Obtener la PILA sin X elemento (ITERATIVO).
4.  Obtener la PILA sin X elemento (RECURSIVO).
5.  Mostrar la PILA actual.
0.  VOLVER AL MENU DEL SISTEMA.

Ingrese el numero de la accion a realizar: 3

sh: 1: cls: not found
Contenido de la pila: 0 3
Que elemento desea eliminar de la PILA?:
```

The screenshot shows the Visual Studio Code editor with a C++ file named `main.c`. The code defines a linked list structure (`PILA`) and a `main` function. The `main` function creates a `PILA` object, adds three elements (1, 2, 3), prints the contents, pauses, and then attempts to free the `Resultado` pointer. A red error message "Exception has occurred. Aborted" is displayed. The terminal window shows the program's execution, including input and output, and ends with a "free(): invalid pointer" error.

```

C main.c > main()
11 int main(){
12     PILA *Resultado = new PILA();
13     Resultado->Agregar(1);
14     Resultado->Agregar(2);
15     Resultado->Agregar(3);
16     Resultado->Mostrar();
17     system("pause");
18     system("cls");
19     free(Resultado);
20 }

```

Exception has occurred. ✕  
Aborted

```

706
707
708 while(bandera != 0);
709

```

OUTPUT TERMINAL PORTS

TERMINAL

```

sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 2

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1

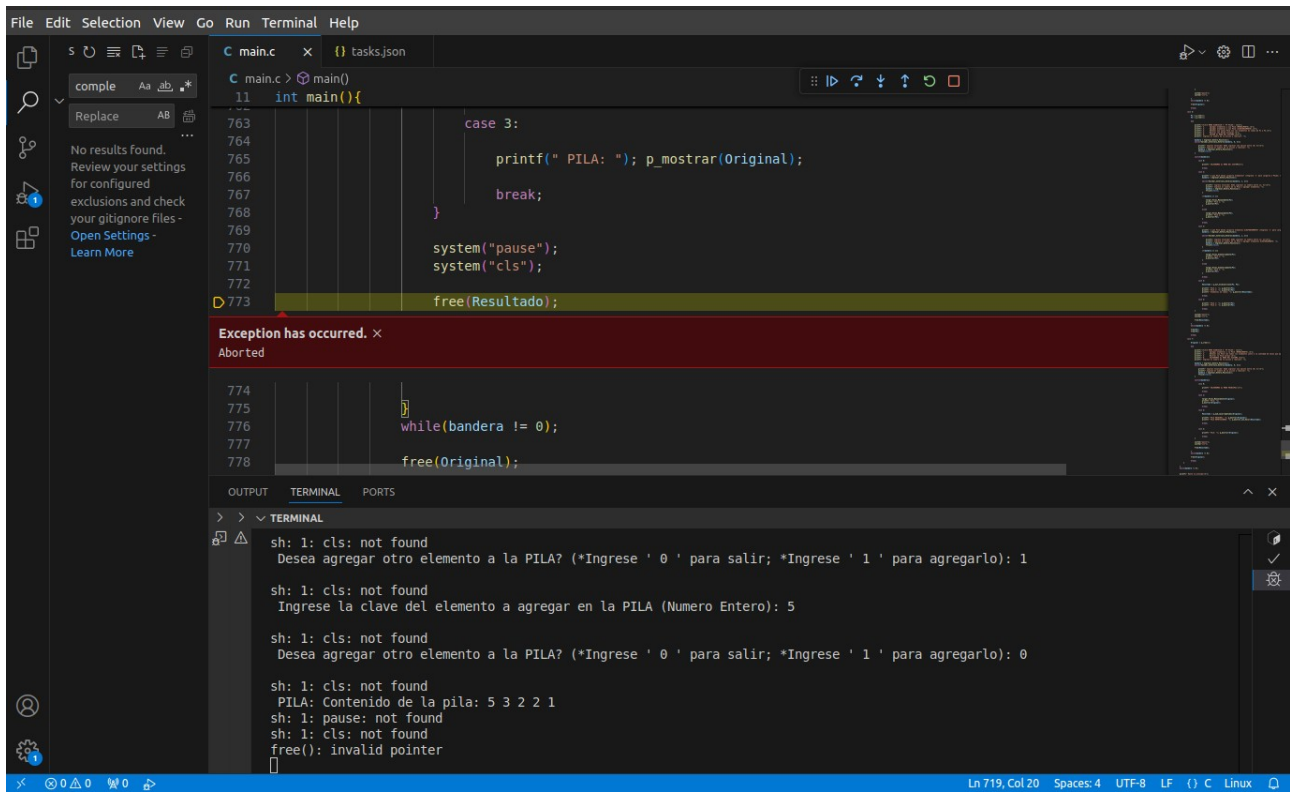
sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 3

sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 0

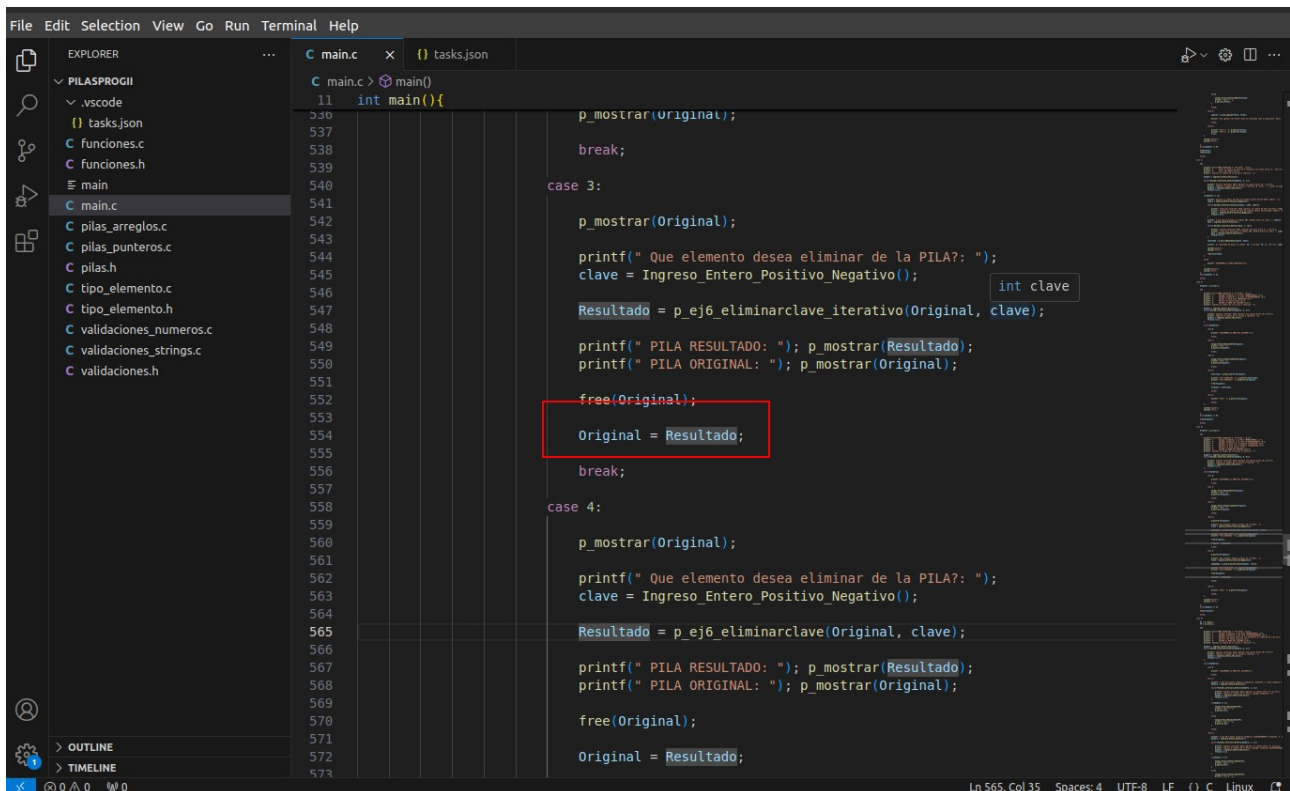
sh: 1: cls: not found
PILA 1: Contenido de la pila: 3 2 1
sh: 1: pause: not found
sh: 1: cls: not found
free(): invalid pointer

```

## GRUPO 2 – Correcciones Trabajo Práctico: PILAS



```
File Edit Selection View Go Run Terminal Help
C main.c x {} tasks.json
C main.c > main()
11 int main(){
763
764
765
766
767
768
769
770
771
772
773 free(Resultado);
Exception has occurred.
Aborted
774
775
776 while(bandera != 0);
777
778 free(Original);
OUTPUT TERMINAL PORTS
sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 1
sh: 1: cls: not found
Ingrese la clave del elemento a agregar en la PILA (Numero Entero): 5
sh: 1: cls: not found
Desea agregar otro elemento a la PILA? (*Ingrese ' 0 ' para salir; *Ingrese ' 1 ' para agregarlo): 0
sh: 1: cls: not found
PILA: Contenido de la pila: 5 3 2 2 1
sh: 1: pause: not found
sh: 1: cls: not found
free(): invalid pointer
Ln 719, Col 20 Spaces: 4 UTF-8 LF C Linux
```



```
File Edit Selection View Go Run Terminal Help
EXPLORER
PILASPROGII
.vscode
{} tasks.json
C funciones.c
C funciones.h
E main
C main.c
C pilas_arreglos.c
C pilas_punteros.c
C pilas.h
C tipo_elemento.c
C tipo_elemento.h
C validaciones_numeros.c
C validaciones_strings.c
C validaciones.h
C main.c x {} tasks.json
C main.c > main()
11 int main(){
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
p_mostrar(Original);
break;
case 3:
p_mostrar(Original);
printf(" Que elemento desea eliminar de la PILA?: ");
clave = Ingreso_Entero_Positivo_Negativo();
Resultado = p_ej6_eliminarclave_iterativo(Original, clave);
printf(" PILA RESULTADO: "); p_mostrar(Resultado);
printf(" PILA ORIGINAL: "); p_mostrar(Original);
free(Original);
Original = Resultado;
break;
case 4:
p_mostrar(Original);
printf(" Que elemento desea eliminar de la PILA?: ");
clave = Ingreso_Entero_Positivo_Negativo();
Resultado = p_ej6_eliminarclave_iterativo(Original, clave);
printf(" PILA RESULTADO: "); p_mostrar(Resultado);
printf(" PILA ORIGINAL: "); p_mostrar(Original);
free(Original);
Original = Resultado;
Ln 565, Col 35 Spaces: 4 UTF-8 LF C Linux
```