

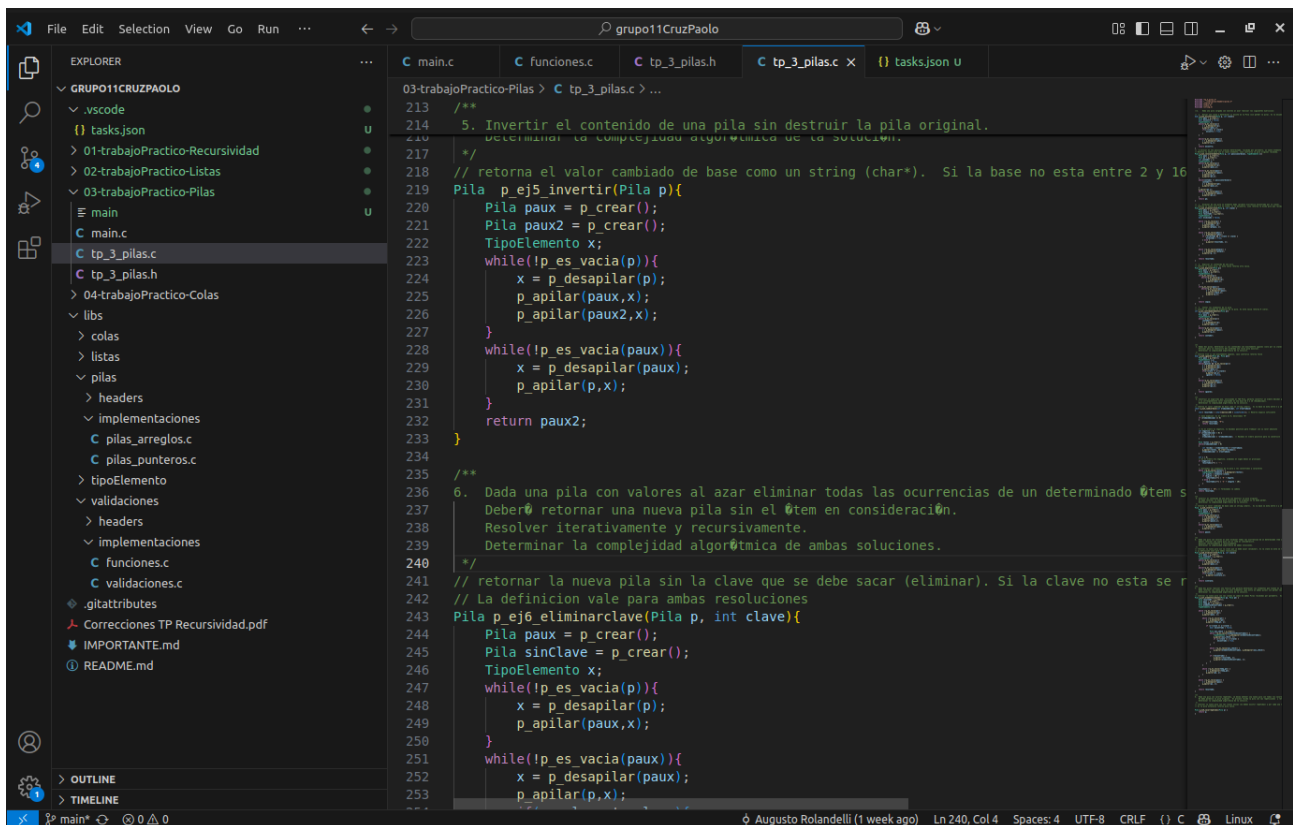
RESULTADO DE LA CORRECCIÓN: **DESAPROBADO**

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.

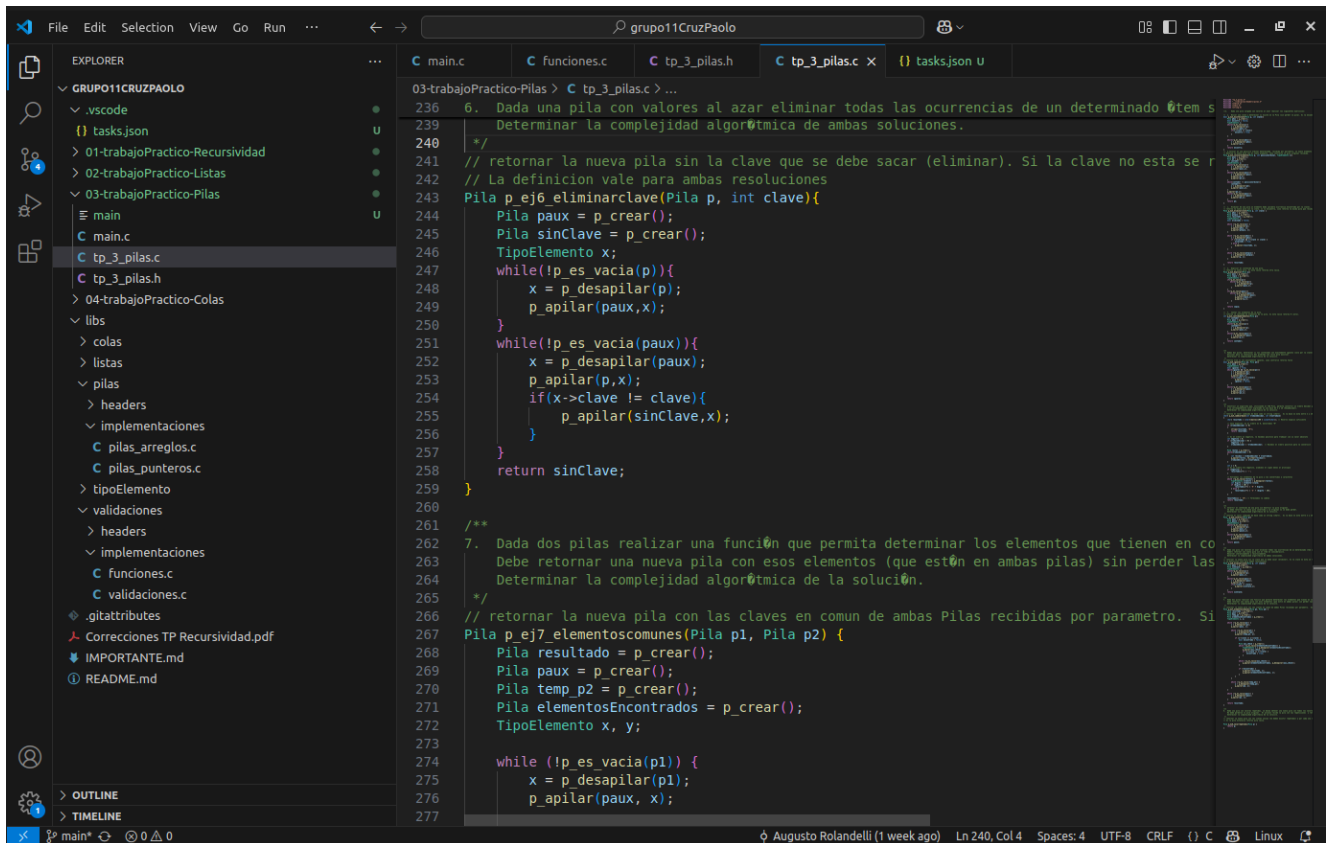
OBSERVACIONES

Ejercicio 6: solo realizan la implementación iterativa.



```
213 /**
214 5. Invertir el contenido de una pila sin destruir la pila original.
215 Determinar la complejidad algorítmica de la solución.
216 */
217 // retorna el valor cambiado de base como un string (char*). Si la base no esta entre 2 y 16
218 Pila p_ej5_invertir(Pila p){
219     Pila paux = p_crear();
220     Pila paux2 = p_crear();
221     TipoElemento x;
222     while(!p_es_vacia(p)){
223         x = p_desapilar(p);
224         p_apilar(paux,x);
225         p_apilar(paux2,x);
226     }
227     while(!p_es_vacia(paux)){
228         x = p_desapilar(paux);
229         p_apilar(p,x);
230     }
231     return paux2;
232 }
233
234 /**
235 6. Dada una pila con valores al azar eliminar todas las ocurrencias de un determinado item.
236 Deberá retornar una nueva pila sin el item en consideración.
237 Resolver iterativamente y recursivamente.
238 Determinar la complejidad algorítmica de ambas soluciones.
239 */
240 // retornar la nueva pila sin la clave que se debe sacar (eliminar). Si la clave no esta se r
241 // La definicion vale para ambas resoluciones
242 Pila p_ej6_eliminarclave(Pila p, int clave){
243     Pila paux = p_crear();
244     Pila sinClave = p_crear();
245     TipoElemento x;
246     while(!p_es_vacia(p)){
247         x = p_desapilar(p);
248         p_apilar(paux,x);
249     }
250     while(!p_es_vacia(paux)){
251         x = p_desapilar(paux);
252         p_apilar(p,x);
253     }
```

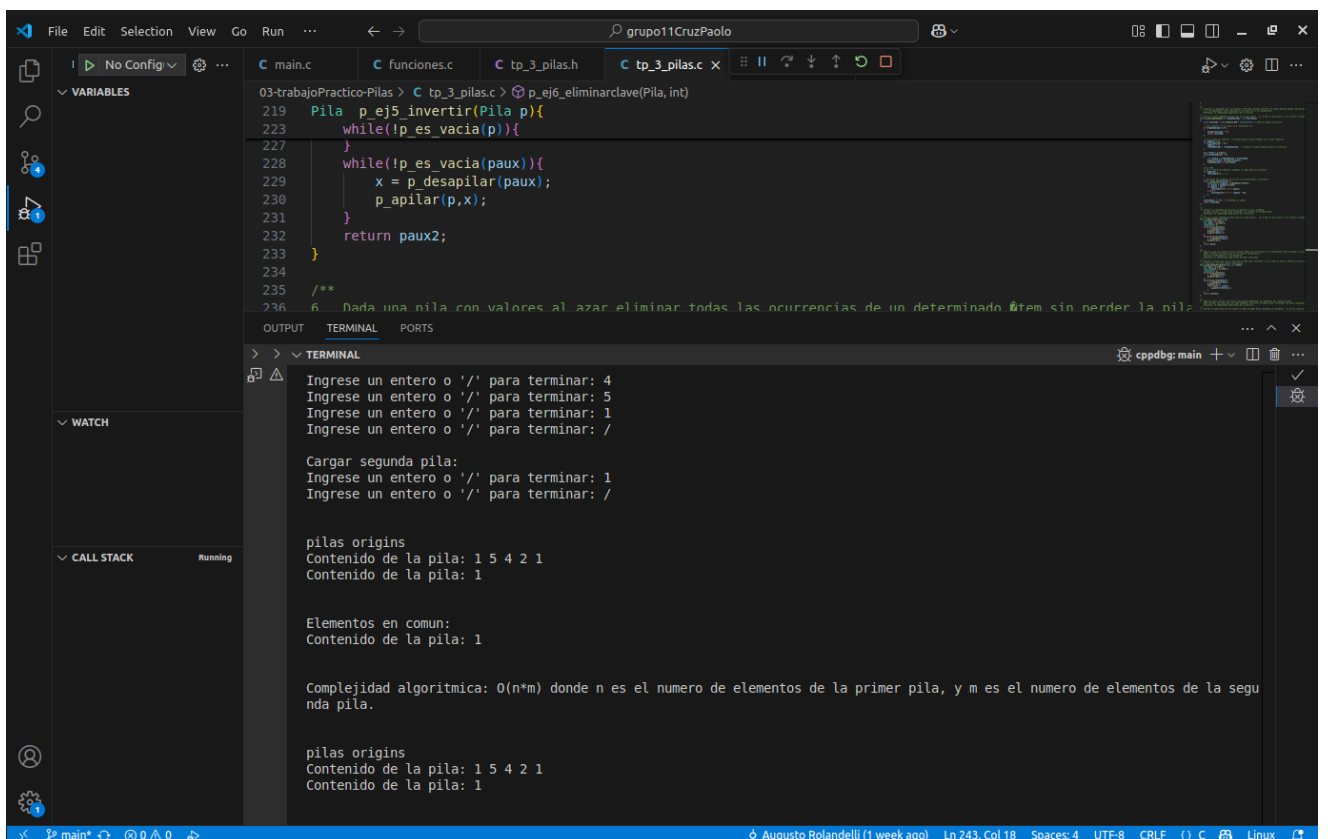
GRUPO 11 – Correcciones Trabajo Práctico: PILAS



```
File Edit Selection View Go Run ... grupo11CruzPaolo
EXPLORER
GRUPO11CRUZPAOLO
  .vscode
  {} tasks.json
  > 01-trabajoPractico-Recursividad
  > 02-trabajoPractico-Listas
  > 03-trabajoPractico-Pilas
    main
    C main.c
    C tp_3_pilas.c
    C tp_3_pilas.h
  > 04-trabajoPractico-Colas
  libs
  > colas
  > listas
  > pilas
    headers
    > implementaciones
      pilas_arreglos.c
      pilas_punteros.c
    > tipoElemento
  > validaciones
    headers
    > implementaciones
      funciones.c
      validaciones.c
  .gitattributes
  Correcciones TP Recursividad.pdf
  IMPORTANTE.md
  README.md
OUTLINE
TIMELINE
main* 0 0 0

03-trabajoPractico-Pilas > C tp_3_pilas.c > ...
236 6. Dada una pila con valores al azar eliminar todas las ocurrencias de un determinado item sin perder la pila.
237 Determinar la complejidad algorítmica de ambas soluciones.
238
239 */
240
241 // retornar la nueva pila sin la clave que se debe sacar (eliminar). Si la clave no esta se r
242 // La definicion vale para ambas resoluciones
243 Pila p_ej6_eliminarclave(Pila p, int clave){
244     Pila paux = p_crear();
245     Pila sinClave = p_crear();
246     TipoElemento x;
247     while(!p_es_vacia(p)){
248         x = p_desapilar(p);
249         p_apilar(paux,x);
250     }
251     while(!p_es_vacia(paux)){
252         x = p_desapilar(paux);
253         p_apilar(p,x);
254         if(x->clave != clave){
255             p_apilar(sinClave,x);
256         }
257     }
258     return sinClave;
259 }
260
261 /**
262 7. Dada dos pilas realizar una función que permita determinar los elementos que tienen en co
263 Debe retornar una nueva pila con esos elementos (que estén en ambas pilas) sin perder las
264 Determinar la complejidad algorítmica de la solución.
265 */
266 // retornar la nueva pila con las claves en comun de ambas Pilas recibidas por parametro. Si
267 Pila p_ej7_elementoscomunes(Pila p1, Pila p2) {
268     Pila resultado = p_crear();
269     Pila paux = p_crear();
270     Pila temp_p2 = p_crear();
271     Pila elementosEncontrados = p_crear();
272     TipoElemento x, y;
273
274     while (!p_es_vacia(p1)) {
275         x = p_desapilar(p1);
276         p_apilar(paux, x);
277     }
```

Ejercicio 7: complejidad incorrecta.



```
File Edit Selection View Go Run ... grupo11CruzPaolo
C main.c C funciones.c C tp_3_pilas.h C tp_3_pilas.c
03-trabajoPractico-Pilas > C tp_3_pilas.c > p_ej6_eliminarclave(Pila, int)
219 Pila p_ej5_invertir(Pila p){
220     while(!p_es_vacia(p)){
221     }
222     while(!p_es_vacia(paux)){
223         x = p_desapilar(paux);
224         p_apilar(p,x);
225     }
226     return paux2;
227 }
228
229 /**
230 6. Dada una pila con valores al azar eliminar todas las ocurrencias de un determinado item sin perder la pila
231 Determinar la complejidad algorítmica de ambas soluciones.
232 */
233
234 // retornar la nueva pila sin la clave que se debe sacar (eliminar). Si la clave no esta se r
235 // La definicion vale para ambas resoluciones
236 Pila p_ej6_eliminarclave(Pila p, int clave){
237     Pila paux = p_crear();
238     Pila sinClave = p_crear();
239     TipoElemento x;
240     while(!p_es_vacia(p)){
241         x = p_desapilar(p);
242         p_apilar(paux,x);
243     }
244     while(!p_es_vacia(paux)){
245         x = p_desapilar(paux);
246         p_apilar(p,x);
247         if(x->clave != clave){
248             p_apilar(sinClave,x);
249         }
250     }
251     return sinClave;
252 }
253
254 /**
255 7. Dada dos pilas realizar una función que permita determinar los elementos que tienen en co
256 Debe retornar una nueva pila con esos elementos (que estén en ambas pilas) sin perder las
257 Determinar la complejidad algorítmica de la solución.
258 */
259 // retornar la nueva pila con las claves en comun de ambas Pilas recibidas por parametro. Si
260 Pila p_ej7_elementoscomunes(Pila p1, Pila p2) {
261     Pila resultado = p_crear();
262     Pila paux = p_crear();
263     Pila temp_p2 = p_crear();
264     Pila elementosEncontrados = p_crear();
265     TipoElemento x, y;
266
267     while (!p_es_vacia(p1)) {
268         x = p_desapilar(p1);
269         p_apilar(paux, x);
270     }
```

OUTPUT TERMINAL PORTS

cpdbg: main + -

```
Ingrese un entero o '/' para terminar: 4
Ingrese un entero o '/' para terminar: 5
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: /

Cargar segunda pila:
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: /

pilas origins
Contenido de la pila: 1 5 4 2 1
Contenido de la pila: 1

Elementos en comun:
Contenido de la pila: 1

Complejidad algorítmica: O(n*m) donde n es el numero de elementos de la primer pila, y m es el numero de elementos de la segun
da pila.

pilas origins
Contenido de la pila: 1 5 4 2 1
Contenido de la pila: 1
```

Ejercicio 8: no lo resuelve.

```

267 Pila p_ej7_elementoscomunes(Pila p1, Pila p2) {
268
269     while (!p_es_vacia(paux)) {
270         x = p_desapilar(paux);
271         p_apilar(p1, x);
272     }
273
274     return resultado;
275 }
276
277 /**
278  * 8. Dada una pila con valores repetidos, se desea obtener una nueva pila con todos los valores (sin repet
279  * No debe perderse la pila original. El proceso recibe la pila con las repeticiones y retorna una nue
280  * Determinar la complejidad algorítmica de la solución.
281  */
282 // retornar la nueva pila con las claves unicas (no deben existir repetidas) y por cada una la cantidad d
283 // Si la pila estuviera vacia retorna pila vacia.
284
285 Pila p_ej8_sacarrepetidos(Pila p) {
286     return p;
287 }
288
289 }
290
291
292

```

OUTPUT TERMINAL PORTS

```

claudiarg@claudia:~/programacion2/2025/TPs Grupos/Correcciones Manuales-Pilas/Grupo 11/grupo11CruzPaolo$
claudiarg@claudia:~/programacion2/2025/TPs Grupos/Correcciones Manuales-Pilas/Grupo 11/grupo11CruzPaolo$

```

```

6 int main() {
7     pilasEjercicio3();
8     break;
9     case 4:
10        pilasEjercicio4();
11        break;

```

OUTPUT TERMINAL PORTS

```

8) Ver cuantas veces se repiten elementos en una pila.
Ingrese el ejercicio al que ingresar ('0' PARA SALIR): 8

Ejercicio de contar elementos repetidos!

Cargar pila:
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: 24
Ingrese un entero o '/' para terminar: 4
Ingrese un entero o '/' para terminar: 7
Ingrese un entero o '/' para terminar: 4
Ingrese un entero o '/' para terminar: 5
Ingrese un entero o '/' para terminar: 2
Ingrese un entero o '/' para terminar: 1
Ingrese un entero o '/' para terminar: 2
La pila esta llena. No se puede apilar mas.
Contenido de la pila: 1 2 5 4 7 4 24 2 1 1

Elementos repetidos:
Contenido de la pila: 1 2 5 4 7 4 24 2 1 1

Complejidad algorítmica: .

Pila original
Contenido de la pila: 1 2 5 4 7 4 24 2 1 1

---MENU---
2) Dada una pila:

```