

APROBADO-

Observaciones generales

Realiza ambas implementaciones. La resolución es genérica. Para la carga aleatoria de datos solicita un tamaño de pila y verifica que sea menor o igual a 100, lo hace por código, no contra la constante definida en la implementación del TAD. Para el procesamiento de la pila toma el valor de la constante definida en el TAD y si la pila cargada tiene mas elementos, muestra el resultado de manera incorrecta. Los datos incorrectos solo son mostrados con la implementación del TAD con arreglos.

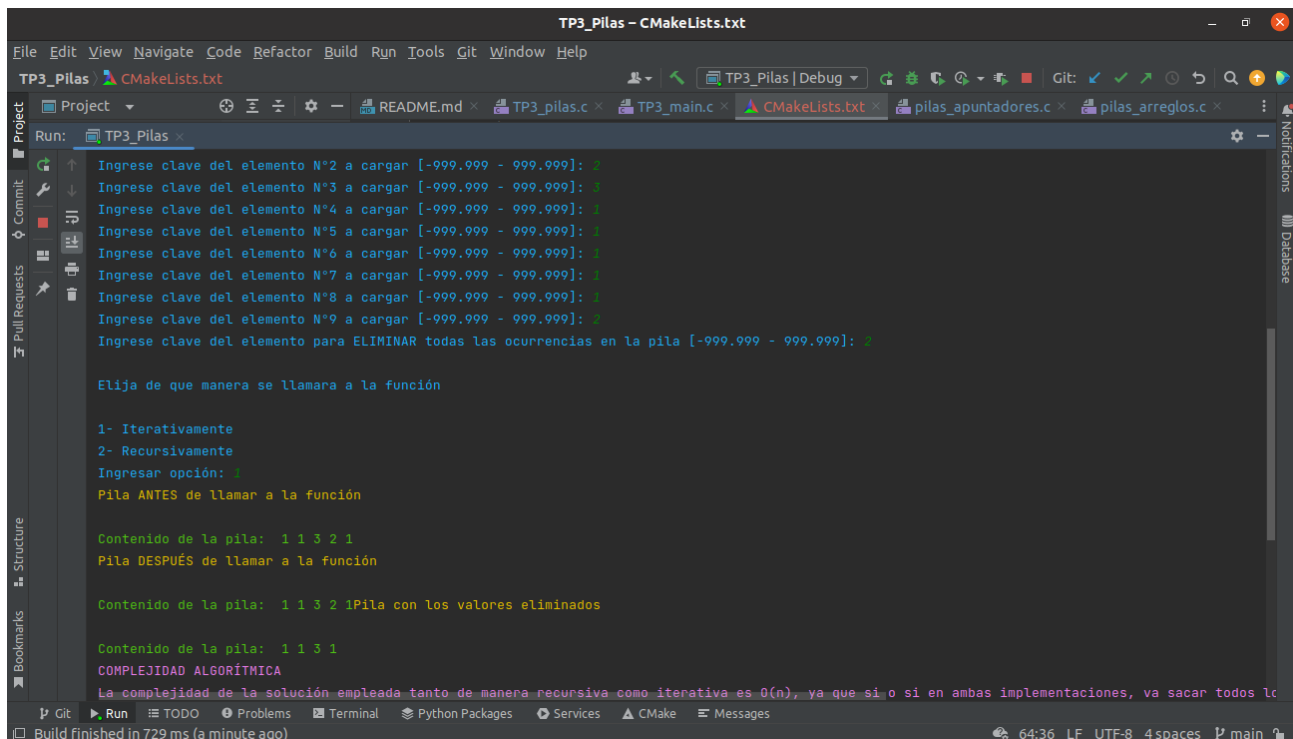
Ejercicio 6 - APROBADO-

Ejercicio 7 – APROBADO-

Muestra repetidos los elementos comunes.

Ejercicio 8 - APROBADO

Responde correctamente sobre complejidad.



```
TP3_Pilas - CMakeLists.txt
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
TP3_Pilas CMakeLists.txt
Run: TP3_Pilas
Ingrese clave del elemento N°2 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°3 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°4 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°5 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°6 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°7 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°8 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°9 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento para ELIMINAR todas las ocurrencias en la pila [-999.999 - 999.999]: 1

Elija de que manera se llamara a la función

1- Iterativamente
2- Recursivamente
Ingresar opción: 1
Pila ANTES de llamar a la función

Contenido de la pila: 1 1 3 2 1
Pila DESPUÉS de llamar a la función

Contenido de la pila: 1 1 3 2 1Pila con los valores eliminados

Contenido de la pila: 1 1 3 1
COMPLEJIDAD ALGORÍTMICA
La complejidad de la solución empleada tanto de manera recursiva como iterativa es O(n), ya que si o si en ambas implementaciones, va sacar todos los
Build finished in 729 ms (a minute ago)
```

TP3_Pilas - CMakeLists.txt

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
TP3_Pilas CMakeLists.txt
Run: TP3_Pilas
Ingrese clave del elemento N°1 a cargar [-999.999 - 999.999]: 1
Ingrese clave del elemento N°2 a cargar [-999.999 - 999.999]: 2
Ingrese clave del elemento N°3 a cargar [-999.999 - 999.999]: 3
Ingrese clave del elemento N°4 a cargar [-999.999 - 999.999]: 4
Ingrese clave del elemento N°5 a cargar [-999.999 - 999.999]: 5
Ingrese clave del elemento N°6 a cargar [-999.999 - 999.999]: 6
Ingrese clave del elemento para ELIMINAR todas las ocurrencias en la pila [-999.999 - 999.999]: 1

Elija de que manera se llamara a la función

1- Iterativamente
2- Recursivamente
Ingresar opción: 1
Pila ANTES de llamar a la función

Contenido de la pila: 2 2 1 1 1
Pila DESPUÉS de llamar a la función

Contenido de la pila: 2 2 1 1 1 Pila con los valores eliminados

Contenido de la pila: 2 2
COMPLEJIDAD ALGORÍTMICA
La complejidad de la solución empleada tanto de manera recursiva como iterativa es O(n), ya que si o si en ambas implementaciones, va sacar todos los elementos de la pila.

=====
```

Build finished in 709 ms (moments ago)

TP3_Pilas - pilas_apuntadores.c

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
TP3_Pilas pilas_apuntadores.c
Project: TP3_Pilas - /programacion2/TPListas/20
  .vscode
  cmake-build-debug
  CMakeLists.txt
  Guia_TP_P2_2023_Pilas.pdf
  pilas.h
  pilas_apuntadores.c
  pilas_arreglos.c
  README.md
Run: TP3_Pilas
Elija de que manera se llamara a la función

1- Iterativamente
2- Recursivamente
Ingresar opción: 1
Pila ANTES de llamar a la función

Contenido de la pila: 2 2 2 2 1 1 1 1
Pila DESPUÉS de llamar a la función

Contenido de la pila: 2 2 2 2 2 1 1 1
Pila con los valores eliminados

Contenido de la pila: 1 1 1 1
```

Build finished in 890 ms (a minute ago)

