

Clase 3

TAD Compuesto: Implementación

Pila y Cola: Implementación

Aplicaciones con TAD Simple y Compuesto

Desarrollo de la clase 3

Repaso TAD Simple: Implementación

TAD Compuesto: Implementación

Repaso Aplicación utilizando TAD compuesto

TAD Pila: Implementación

TAD Cola: Implementación

Revisar soluciones de TP 1

Realizar ejercicios TP2

Re pa so

Ejemplo Implementación TAD Simple

```
# TAD Libro
```

```
#libro=["", "", "", 0]
```

```
#def crearLib():  
    #Crea un libro vacio  
    libro=["", "", "", 0]  
    return libro
```

```
#def cargarLib(libro,n,e,a,p):  
    #Carga los datos de un libro:  
    # Nom, Edit, Autor, Precio  
    libro[0]=n  
    libro[1]=e  
    libro[2]=a  
    libro[3]=p
```

```
#def verNom(libro):  
    #Retorna el nombre de un libro  
    return libro[0]
```

```
#def verEdit(libro):  
    #Retorna la editorial de un libro  
    return libro[1]
```

```
#def verAutor(libro):  
    #Retorna el autor de un libro  
    return libro[2]
```

```
#def verPre(libro):  
    #Retorna el autor de un libro  
    return libro[3]
```

Re pa so

Ejemplo Implementación TAD Simple

```
#def modNom(libro,n):  
    #Modifica el nombre de un libro  
    libro[0]=n  
  
#def modEdit(libro,e):  
    #Modifica la editorial de un libro  
    libro[1]=e  
  
#def modAutor(libro,a):  
    #Modifica el autor de un libro  
    libro[2]=a  
  
#def modPre(libro,p):  
    #Modifica el autor de un libro  
    libro[3]=p
```

```
#def asignarLib(libro1,libro2):  
    #Asigna datos de un libro en otro  
    modNom(libro1,verNom(libro2))  
    modEdit(libro1,verEdit(libro2))  
    modAutor(libro1,verAutor(libro2))  
    modPre(libro1,verPre(libro2))
```

Ejemplo Implementación TAD Compuesto

```
# TAD Libreria
```

```
#def crearlibreria():  
    #Crea una libreria vacia  
    libreria=[]  
    return libreria  
  
#def agregarLibro(libreria,l):  
    #Agrega un libro a la librería  
    libreria.append(l)  
  
#def eliminarLibro(libreria,l):  
    #Elimina un libro de la librería  
    libreria.remove(l)  
  
#def recuperarLibro(libreria,i):  
    #Retorna el libro de la posicion iesima  
    return libreria[i-1]  
  
#def tamaño(libreria):  
    #Retorna la cantidad de libros de la librería  
    return len(libreria)
```

Re pa so

Ejemplo Aplicación utilizando TAD Compuesto

- Cargar 4 libros a una librería
- Imprimir los datos de todos los libros de la librería

```
import TadLibro, TadLibreria

from TadLibro import *
from TadLibreria import *

#Crea la libreria
lib=crearLibreria()

#Cargar la Libreria
for i in range(1,4):
    #Crea y carga los datos del libro1
    l=crearLib()
    n=input("Ingrese un nombre")
    e=input("Ingrese un editorial")
    a=input("Ingrese un autor")
    p=float(input("Ingrese un precio"))
    cargarLib(l,n,e,a,p)
    agregarLibro(Lib,l)
```

```
#Imprime los datos de la libreria
print ("Imprime los datos de la libreria")
for i in range(0,tamano(lib)):
    l=recuperarLibro(lib,i)
    print ("Imprime los datos de un libro")
    print (verNom(l))
    print (verEdit(l))
    print (verAutor(l))
    print (verPre(l))
```

Re pa so

Ejemplo Aplicación utilizando TAD Compuesto

- Recuperar e imprimir el segundo libro
- Eliminar el libro recuperado e imprimir los datos de todos los libros de la libreria

```
#Recupera e imprime el segundo libro
print ("Imprime datos del libro recup")
```

```
l=recuperarLibro(lib,1)
```

```
print ("Imprime los datos de un libro")
print (verNom(l))
print (verEdit(l))
print (verAutor(l))
print (verPre(l))
```

```
print ("Elimina el libro recuperado")
#Elimina el libro recuperado
```

```
eliminarLibro(lib,1)
```

```
#Imprime libros que quedan en la libreria
print ("Imprime datos de libros de libreria")
```

```
for i in range(0,tamano(lib)):
    l=recuperarLibro(lib,i)
    print ("Imprime los datos de un libro")
    print (verNom(l))
    print (verEdit(l))
    print (verAutor(l))
    print (verPre(l))
```

Ejemplo Implementación TAD Pila

```
def crearPila():  
    #Crea una pila vacía  
    pila=[]  
    return pila  
  
def esVacia(pila):  
    #Retorna Verdadero si la pila no tiene elementos  
    return len(pila)==0  
  
def apilar(pila,elem):  
    #Agrega un elemento al final de la pila  
    pila.append(elem)  
  
def desapilar(pila):  
    #Retorna y elimina el ultimo elemento de la pila  
    elem=pila[len(pila)-1]  
    pila.pop()  
    return elem
```

```
def tamaño(pila):  
    #Retorna la cantidad de elementos de la pila  
    return len(pila)  
  
def copiarPila(pila1,pila2):  
    aux=crearPila()  
    while not esVacia(pila2):  
        elem=desapilar(pila2)  
        apilar(aux,elem)  
    while not esVacia(pila1):  
        elem=desapilar(pila1)  
        apilar(pila2,elem)
```


Ejemplo Implementación TAD Cola

```
def crearCola():
```

```
    #Crea una cola vacia
```

```
    cola=[]
```

```
    return cola
```

```
def esVacia(cola):
```

```
    #Retorna Verdadero si la cola no tiene elementos
```

```
    return len(cola)==0
```

```
def encolar(cola,elem):
```

```
    #Agrega un elemento al final de la cola
```

```
    cola.append(elem)
```

```
def desencolar(cola):
```

```
    #Retorna y elimina el primer elemento de la cola
```

```
    elem=cola[0]
```

```
    del cola[0]
```

```
    return elem
```

```
def tamanio(cola):
```

```
    #Retorna la cantidad de elementos de la cola
```

```
    return len(cola)
```

```
def copiarCola(cola1,cola2):
```

```
    aux=crearCola()
```

```
    while not esVacia(cola2):
```

```
        elem=desencolar(cola2)
```

```
        encolar(aux,elem)
```

```
    while not esVacia(aux):
```

```
        elem=desencolar(aux)
```

```
        encolar(cola1,elem)
```

```
        encolar(cola2,elem)
```

Ejercicios Resueltos TP 1

Revisar resolución de ejercicios TP 1

Ejercicios para resolver

Resolver los ejercicios 1 y 2 del TP 2