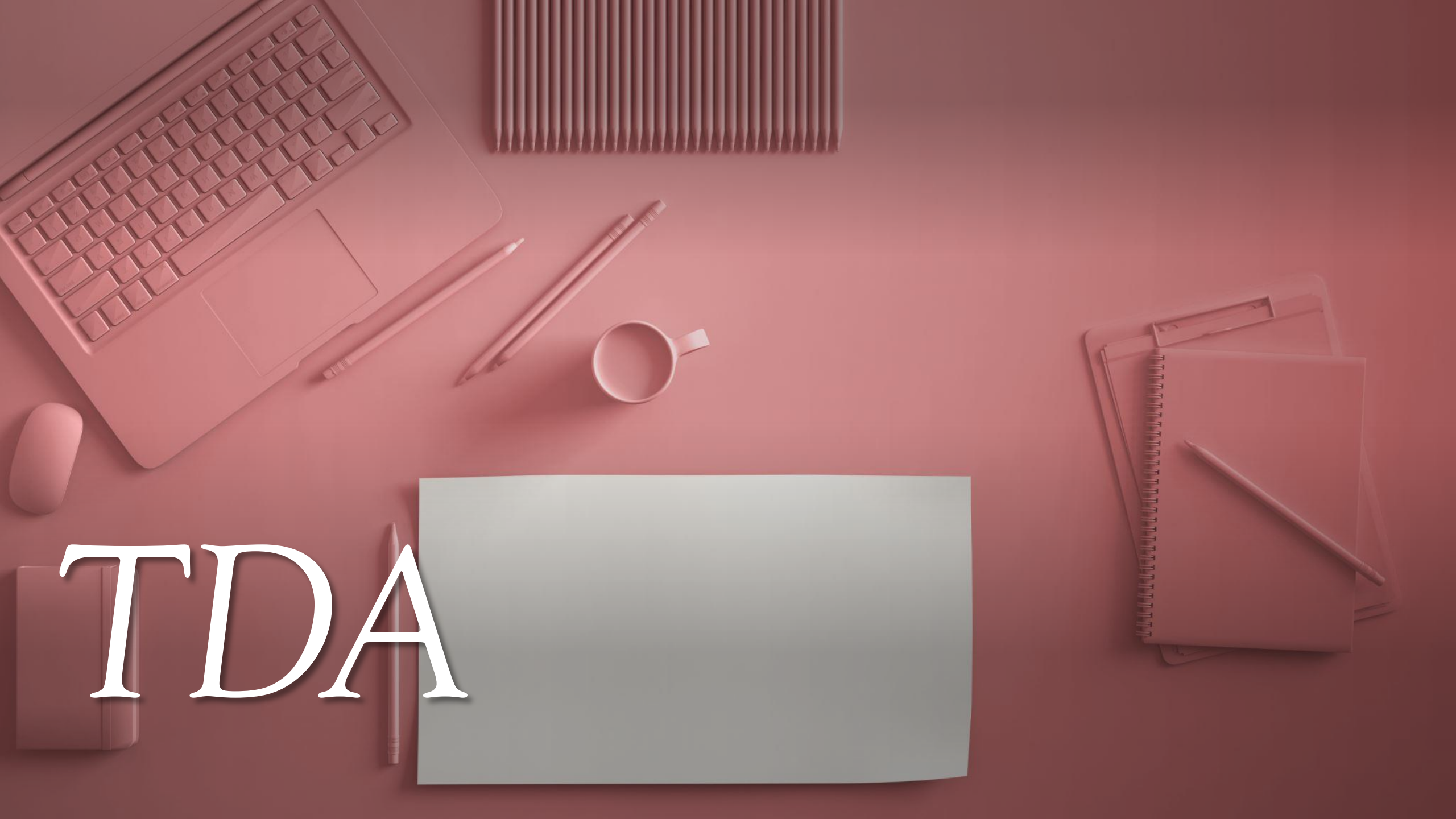
Programación II
Repaso parcial

2C 2023 TN – Ing. Elizabeth Barrera

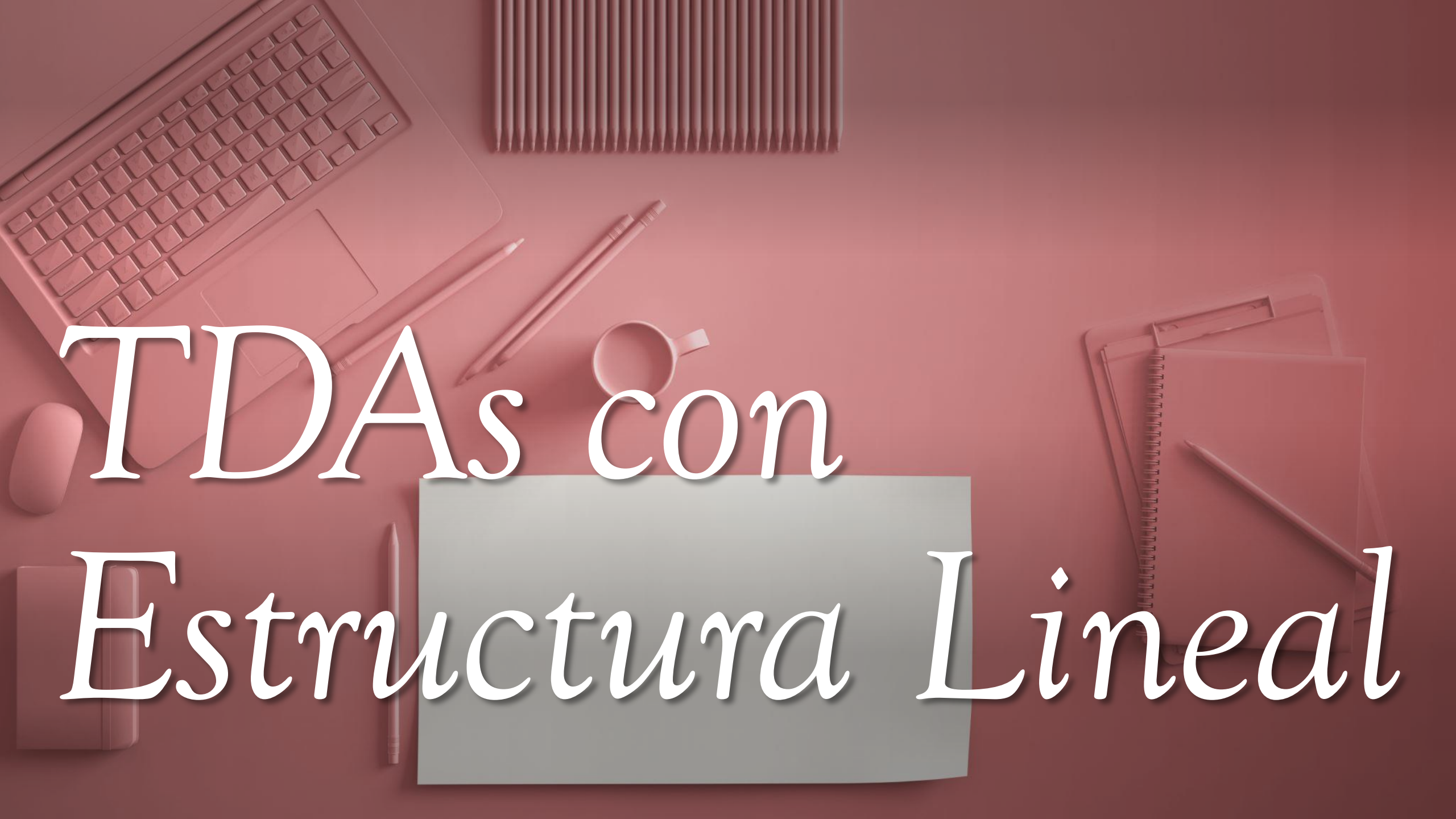
Temas

- *TDA*
- *TDAs con estructura lineal*
- *Análisis de costos*
- *Aclaraciones*
- *Ejemplos*


TDA

TDA

- *Es una **abstracción**, ignoramos algunos detalles y nos concentramos en los que nos interesan.*
- *Existen siempre **2 visiones** diferentes en el TDA: usuario e implementador. Son separadas, y una oculta a la otra.*
- *A la definición del TDA la llamamos **especificación**, es única. Representación en Java: interfaz.*
- *A la forma de llevar a cabo lo definido lo denominamos **implementación**. Pueden existir diversas y son intercambiables (transparente al usuario). Existen estáticas y dinámicas (según las estructuras usadas). Representación en Java: clase que implementa la interfaz.*
- *Las **precondiciones** son condiciones que deben cumplirse antes de la ejecución de la operación.*


TDAs con
Estructura Lineal

TDAs con estructura lineal

- Estructuras que permiten almacenar valores, eliminarlos y recuperarlos.
- Se debe utilizar la más apropiada (según sus características) para cada caso particular.
- Si se destruyen al utilizarlas y las necesitamos luego, se debe hacer una copia previa.

Características

	Ordenamiento	Accesos	Particularidades	Recorrido
Pilas	Por orden de llegada	A un único elemento (tope)	Estructura LIFO	Hasta vaciarla, desapilando Se destruye la pila
Colas	Por orden de llegada	A un único elemento (primero)	Estructura FIFO	Hasta vaciarla, desacolando Se destruye la cola
Colas con prioridad	Según prioridad	A un único elemento (primero)	Existe una prioridad	Hasta vaciarla, desacolando Se destruye la cola
Conjuntos	Sin orden	A cualquier elemento	Sin orden, sin duplicados	Hasta vaciarlo, sacando elementos elegidos (guardar elemento, no volver a elegir) Se destruye el conjunto
Diccionarios simples	Sin orden	A cualquier elemento	Valor recuperable con la clave	Con el conjunto de claves, hasta vaciarlo No se destruye el diccionario (sí el conjunto)
Diccionarios múltiples	Sin orden	A cualquier elemento	Valores recuperables con la clave	Con el conjunto de claves, hasta vaciarlo No se destruye el diccionario (sí el conjunto)

Implementaciones Estáticas

Estrategias

- Se utiliza un ***arreglo*** para guardar los datos.
- Se utiliza un ***entero*** que define la cantidad de elementos guardados (el resto son basura).
- Si se necesita guardar varios datos, se usan estructuras (para la cola con prioridad y diccionarios); se accede a sus elementos internos mediante la notación de punto.
- ***Agregado*** sencillo: en la primera posición libre (a donde apunta el entero).
- ***Eliminado*** mediante borrado lógico (los elementos ubicados desde la posición que apunta el entero en adelante no pertenecen al TDA).
- Eliminado con orden: corrimiento.
- Eliminado sin orden: pisado con el último elemento.
- El arreglo, el entero y todo método privado (que no exista en la especificación) son ***inaccesibles por el usuario*** (sólo se pueden utilizar dentro de la implementación).


Implementaciones
Dinámicas

Estrategias

- *Se utilizan **listas** de nodos enlazados, con referencia al primer elemento.*
- *Los **nodos** contienen un valor y una referencia al siguiente nodo, se pueden modificar a necesidad.*
- *Se debe **recorrer** la estructura siempre que se desee acceder a otro elemento que no sea el primero, utilizando un nodo auxiliar (para evitar perder la referencia al primero).*
- *Se agregan o eliminan nodos al necesitarlos o dejar de necesitarlos.*
- *Se pueden utilizar otras referencias, de ser necesarias (como en las colas).*
- *Se pueden **agregar** elementos al principio (la más sencilla), al final o de forma ordenada, según corresponda.*
- *Para **eliminar** un nodo se lo deja inaccesible circunvalándolo, apuntando el anterior al siguiente; si es el primer nodo debe desplazarse la referencia de la lista al segundo nodo.*

Análisis de Costos

Análisis de costos

- *Análisis de **complejidad temporal**: cantidad de operaciones llevadas a cabo.*
- *Existen 3 criterios de **clasificación**:*
  - ***Costo constante** (C): el costo no depende de la cantidad de elementos de la estructura. Las operaciones elementales tienen este costo.*
  - ***Costo lineal** (L): el costo depende linealmente de la cantidad de elementos de la estructura. Un ciclo que recorre la estructura (dentro del cual se poseen operaciones elementales) tiene este costo.*
  - ***Costo polinómico** (P): el costo es mayor a lineal, puede ser cuadrático, cúbico, etc. Los ciclos anidados que recorren la estructura tienen este costo.*

Comparación

Implementaciones estáticas

Pila	
inicializarPila	constante
apilar	constante
desapilar	constante
tope	constante
pilaVacia	constante

Cola	
inicializarCola	constante
acolar	lineal
desacolar	constante
primero	constante
colaVacia	constante

Cola con prioridad	
inicializarCola	constante
acolarPrioridad	lineal
desacolar	constante
primero	constante
prioridad	constante
colaVacia	constante

Conjunto	
inicializarConjunto	constante
agregar	lineal
sacar	lineal
elegir	constante
pertenece	lineal
conjuntoVacio	constante

Diccionario simple	
inicializarDiccionario	constante
agregar	lineal
eliminar	lineal
recuperar	lineal
claves	polinómica

Diccionario múltiple	
inicializarDiccionario	constante
agregar	lineal
eliminar	lineal
eliminarValor	lineal
recuperar	polinómica
claves	polinómica

Implementaciones dinámicas

Pila	
inicializarPila	constante
apilar	constante
desapilar	constante
tope	constante
pilaVacia	constante

Cola	
inicializarCola	constante
acolar	constante
desacolar	constante
primero	constante
colaVacia	constante

Cola con prioridad	
inicializarCola	constante
acolarPrioridad	lineal
desacolar	constante
primero	constante
prioridad	constante
colaVacia	constante

Conjunto	
inicializarConjunto	constante
agregar	lineal
sacar	lineal
elegir	constante
pertenece	lineal
conjuntoVacio	constante

Diccionario simple	
inicializarDiccionario	constante
agregar	lineal
eliminar	lineal
recuperar	lineal
claves	polinómica

Diccionario múltiple	
inicializarDiccionario	constante
agregar	lineal
eliminar	lineal
eliminarValor	lineal
recuperar	polinómica
claves	polinómica


Aclaraciones

Aclaraciones

- *Leer con detalle el enunciado antes de comenzar.*
- *Antes de empezar, determinar si el ejercicio está del lado del usuario o del implementador.*
- *Darle suma importancia a definir la estrategia.*
- *Pueden existir preguntas teóricas sencillas.*
- *El uso de sentencias "break" y "continue" no está permitido, así como tampoco de colecciones de Java.*


Ejemplos

Ejemplos

- *Realizar el método externo clavesOrdenadas que muestre en forma ordenada las claves de un diccionario múltiple que recibe por parámetro.*

Ejemplos

- *Realizar la implementación de un nuevo TDA similar a ColaPrioridadTDA, en el cual se contará con los mismos métodos que ColaPrioridadTDA además del método interno sumaPrioridades. El mismo debe devolver la suma de todas las prioridades que estén guardadas.*

Ejemplos

- *Realizar el método externo sonDiccionariosIguales que verifique si son iguales dos DiccionarioSimpleTDA que se reciben por parámetro.*

Ejemplos

- *Realizar la implementación de un nuevo TDA ConjuntoDobleTDA, en el cual se permite una repetición de cada elemento, o sea cada elemento puede estar una o dos veces. Se agrega sólo si no existe o si está una única vez (sino, no se hace nada). Se elimina una única vez (si el elemento no está, no se hace nada; si está una vez, el elemento deja de pertenecer al conjunto; si está dos veces, pasa a quedar una única vez). La pertenencia devuelve la cantidad de veces que está el elemento (0, 1 o 2).*

Especificación - Interfaz

```
public interface ConjuntoDobleTDA {
    void inicializarConjunto( );
    void agregar(int x);
    void sacar(int x);
    int elegir( );
    int perteneceCant(int x);
    boolean conjuntoVacio( );
}
```


¡Muchas Gracias!

Bibliografía

- *Programación II – Apuntes de Cátedra – V1.3 – Cuadrado Trutner – UADE*
- *Programación II – Apuntes de Cátedra – Wehbe – UADE*