Programación II
Tipos de Datos Abstractos

2C 2023 – Ing. Elizabeth Barrera

Temas

- *TDA*
- *Especificación de TDA*
- *Implementación de TDA*

Tipos de Datos Abstractos - TDA

TDA

*En un tipo de dato abstracto el concepto de **abstracción** está asociado al modo en que se los definen.*

*Este **modelo** estará formado por las operaciones que tiene y el comportamiento asociado a esas operaciones, es decir, estará centrado en qué hace el TDA y no cómo lo hace (la abstracción significa el ocultamiento del cómo), o viceversa.*

TDA

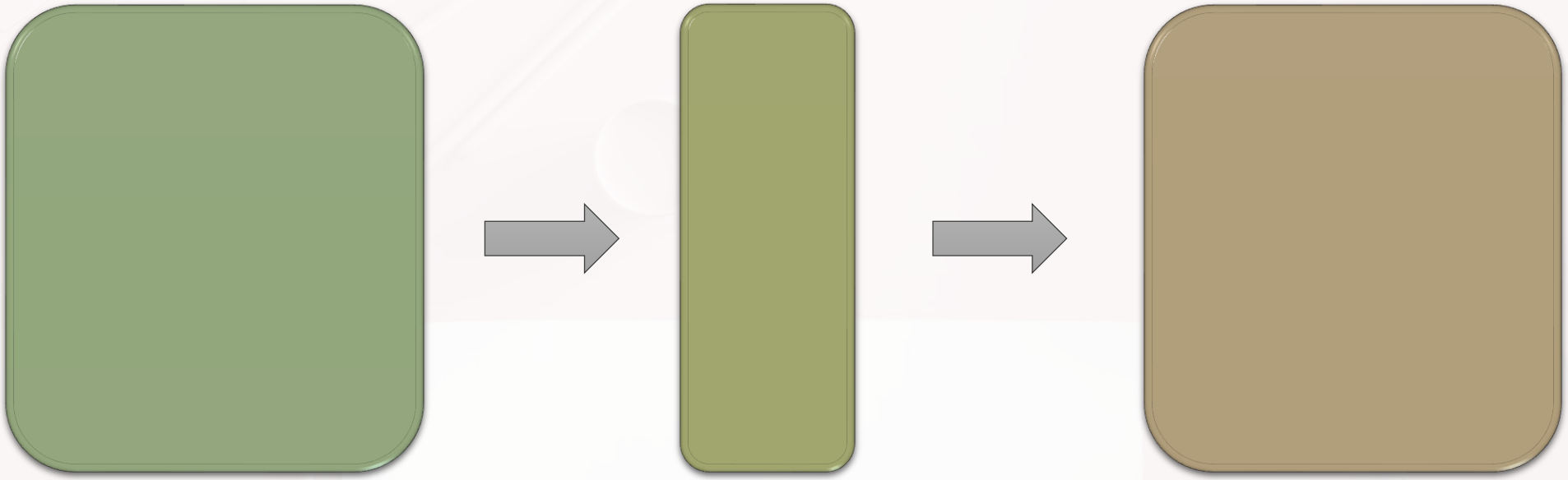
- *Vamos a **separar** qué hace un TDA de cómo lo hace, vamos a definirlos y utilizarlos sin necesidad de conocer cómo se hacen cada una de las operaciones.*
- *Por qué? Es más fácil tratar con lo general, que tener en cuenta detalles que pueden no ser importantes. Un TDA es una **abstracción**, ignoramos algunos detalles y nos concentramos en los que nos interesan.*

TDA

- *A la definición del TDA la llamaremos **especificación** y a la forma de llevar a cabo lo definido por un TDA lo denominaremos **implementación**.*
- *Por lo tanto, para un mismo TDA vamos a tener diferentes implementaciones.*

Recordar que:
*Existen siempre 2 visiones diferentes en el TDA: **usuario** e **implementador**. Son separadas, y una oculta a la otra.*

TDA



Clase que ***utiliza*** el TDA

Interfaz

Clase que ***implementa*** el TDA

Especificación de TDA

Especificación

- En Java utilizaremos la notación de ***interfaces*** para especificar TDA.
- Para especificar una operación de un TDA utilizaremos un ***prototipo de un método*** en Java, incluye:
  - el nombre de la operación
  - los parámetros que recibe con su tipo
  - el tipo de dato que retornan
  - un comentario de lo que hace la operación y sus precondiciones

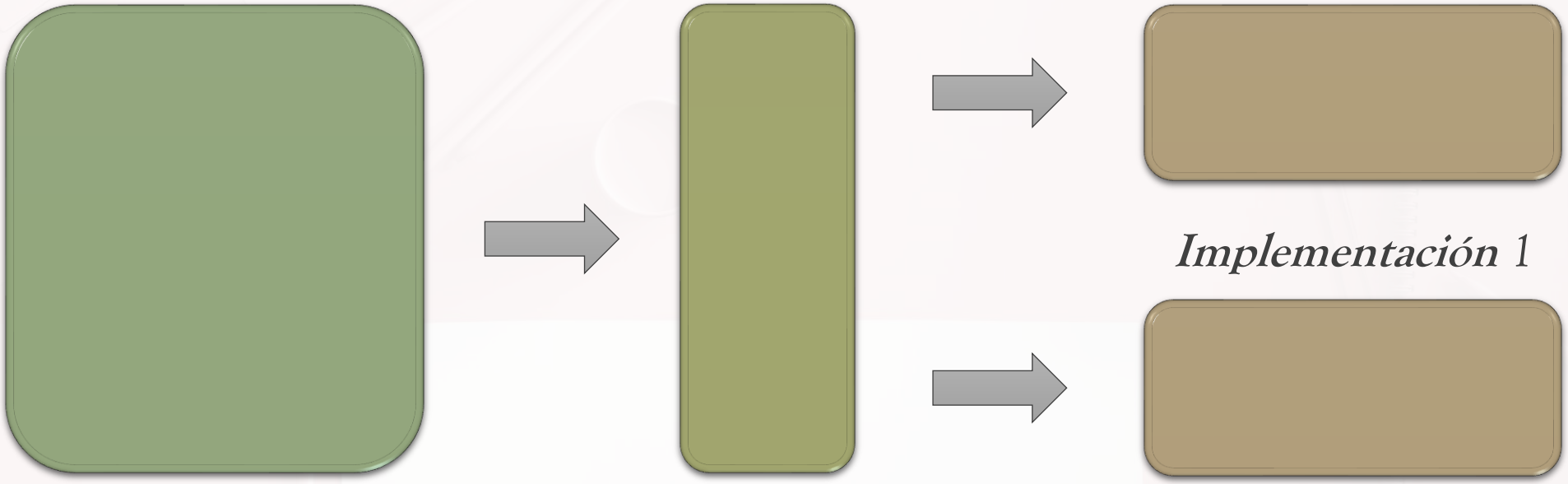
Especificación - Interfaz

```
public interface EjemploTDA {
    // descripción de la acción
    // que realiza el método
    int metodo1(int a, int b);
}
```

Especificación - Por qué?

- *Para el usuario no es necesario conocer los **detalles** de la implementación. Basta con conocer la semántica de los métodos de la interfaz.*
- *Así una implementación podría **intercambiarse** por otra sin que el usuario pudiera percibir la diferencia desde el punto de vista de la semántica.*

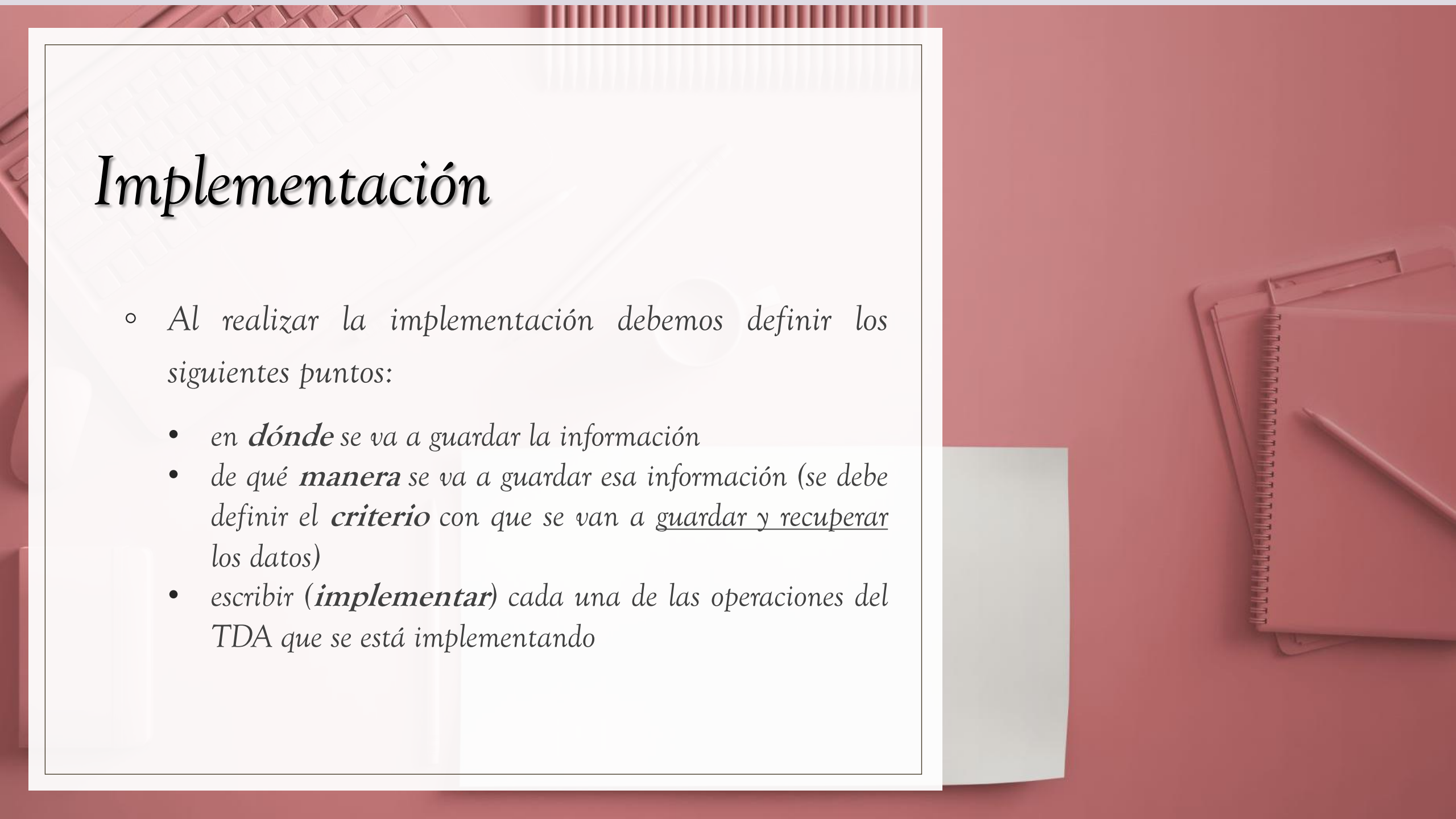
TDA



Implementación de TDA

Implementación

- *Al realizar la implementación debemos definir los siguientes puntos:*
  - *en **dónde** se va a guardar la información*
  - *de qué **manera** se va a guardar esa información (se debe definir el **criterio** con que se van a guardar y recuperar los datos)*
  - *escribir (**implementar**) cada una de las operaciones del TDA que se está implementando*



Implementación - posibilidades

- *En general consideraremos dos posibilidades de implementación: estática y dinámica.*
- *En las **implementaciones estáticas**, los datos se guardan en arreglos, es decir tendremos que definir la capacidad en el momento de inicializar las estructuras.*
- *En las **implementaciones dinámicas**, los datos se guardan en estructuras enlazadas dinámicas, es decir que se van creando según hacen falta.*

Implementación

- *Para implementar un TDA en Java, vamos a escribir una clase, a la que le indicaremos que es la implementación de un TDA con la palabra reservada* ***implements****.*
- *Luego definiremos la* ***estrategia*** *a utilizar, a continuación la estructura de* ***datos****, y por último,* ***codificaremos*** *cada uno de los métodos especificados.*
- *Si fuera necesario, se podrán agregar métodos auxiliares, que serán internos a la clase.*

Recordar que:
Para una misma especificación de TDA pueden existir diferentes implementaciones.

Implementación

```
public class ClaseEjemplo
implements EjemploTDA {

    ...
}
```


¡Muchas Gracias!

Bibliografía

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