Programación II
Conjuntos

2C 2023 TN – Ing. Elizabeth Barrera

Temas

- *TDA*
- *Conjuntos*
- *Especificación*
- *Ejemplos*
- *Implementación estática*

TDA

- *Es una **abstracción**, ignoramos algunos detalles y nos concentramos en los que nos interesan.*
- *A la definición del TDA la llamamos **especificación** y a la forma de llevar a cabo lo definido lo denominamos **implementación**.*

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


Conjuntos

Conjuntos

Un conjunto es una colección de elementos en la que ***no existen duplicados***, ***ni un orden*** *particular.*

Se nos permite conocer si un elemento dado pertenece a la estructura.

Un conjunto puede verse como una bolsa de elementos.

Especificación

*Las operaciones que necesitaremos son: agregar y eliminar datos del conjunto (que llamaremos **agregar** y **sacar**), elegir un elemento arbitrario del conjunto (que llamaremos **elegir**), consultar si el conjunto está o no vacío (que llamaremos **conjuntoVacio**) y si un elemento dado pertenece o no al conjunto (que llamaremos **pertenece**). A estas operaciones agregaremos la inicialización de un conjunto (que llamaremos **inicializarConjunto**).*

Especificación - Operaciones

- ***inicializarConjunto***: *permite inicializar la estructura del conjunto.*
- ***agregar***: *permite agregar un elemento al conjunto (se supone que el conjunto está inicializado).*
- ***sacar***: *permite eliminar del conjunto un elemento dado (se supone que el conjunto está inicializado).*
- ***elegir***: *devuelve un elemento cualquiera del conjunto (se supone que el conjunto está inicializado y no está vacío).*

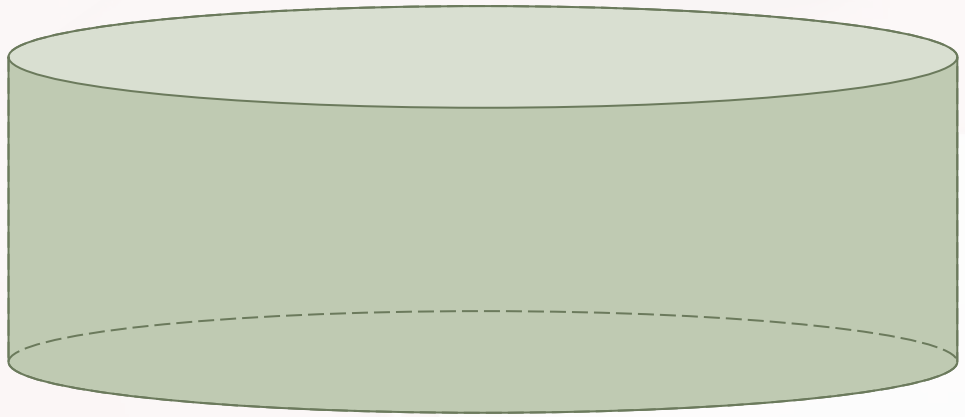
Recordar que:
*Las **precondiciones**, son condiciones que deben cumplirse antes de la ejecución de la operación.*

Especificación - Operaciones

- ***pertenece***: *permite conocer si un elemento dado se encuentra en el conjunto (se supone que el conjunto está inicializado).*
- ***conjuntoVacio***: *indica si el conjunto contiene elementos o no (se supone que el conjunto está inicializado).*

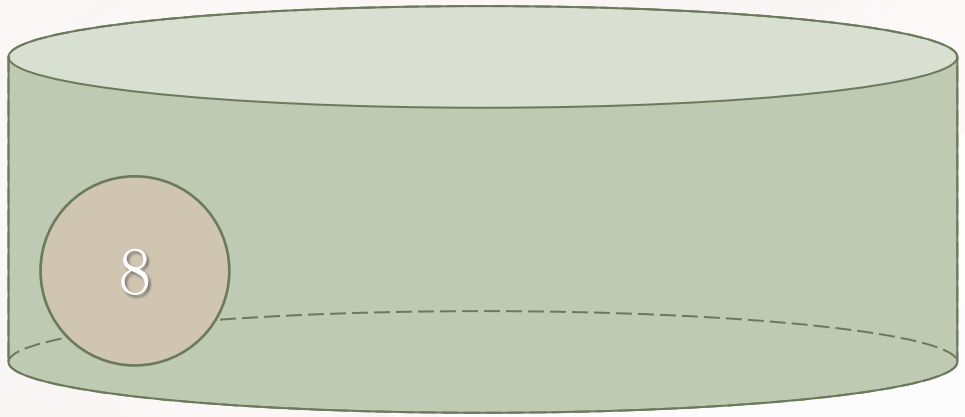
Recordar que:
*Las **precondiciones**, son condiciones que deben cumplirse antes de la ejecución de la operación.*

Conjuntos - Especificación



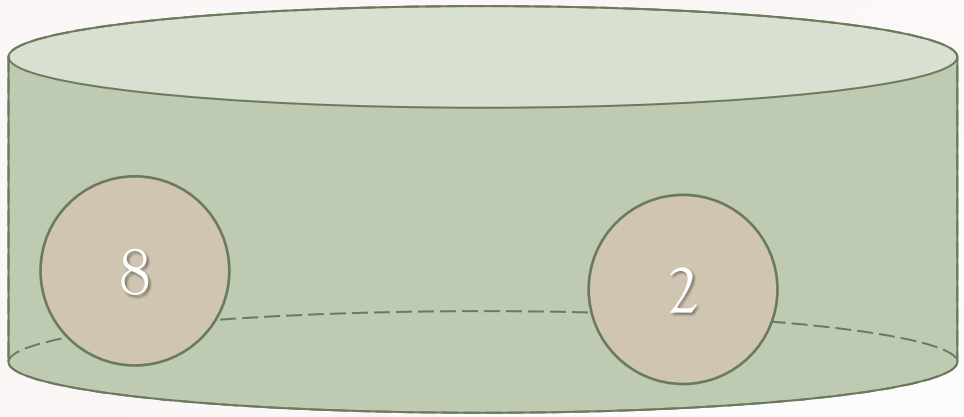
conjuntoVacio () = true

Conjuntos - Especificación



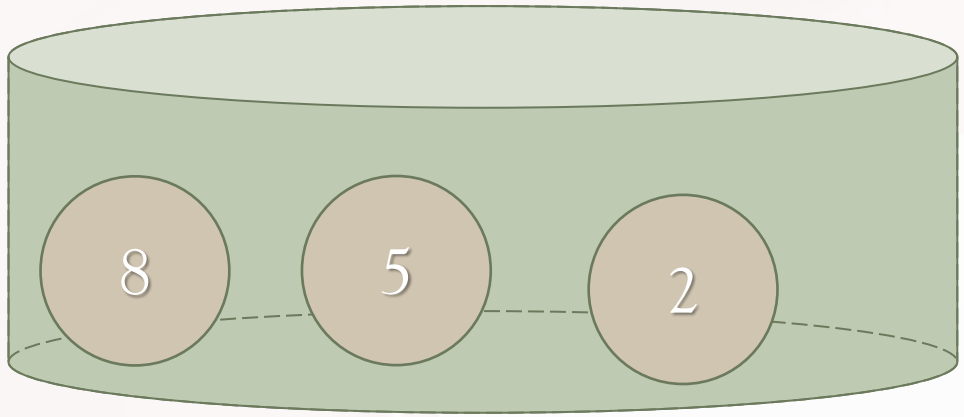
agregar (8)
conjuntoVacio () = false
pertenece (2) = false
pertenece (5) = false

Conjuntos - Especificación



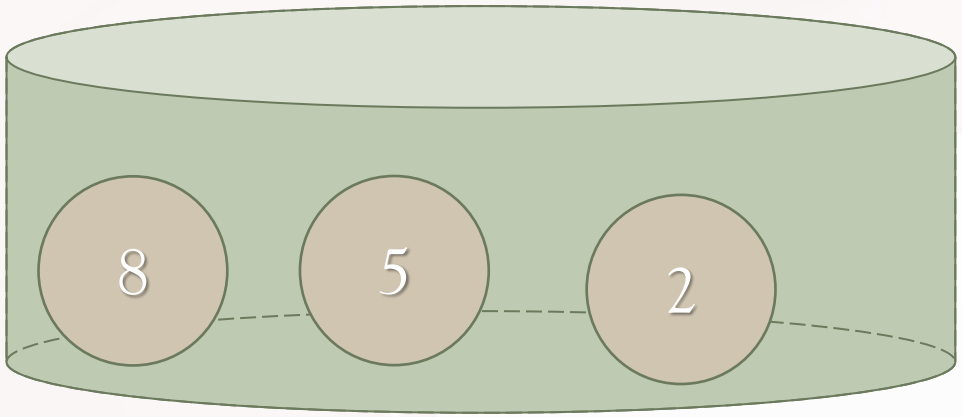
agregar (2)
conjuntoVacio () = false
pertenece (2) = true
pertenece (5) = false

Conjuntos - Especificación



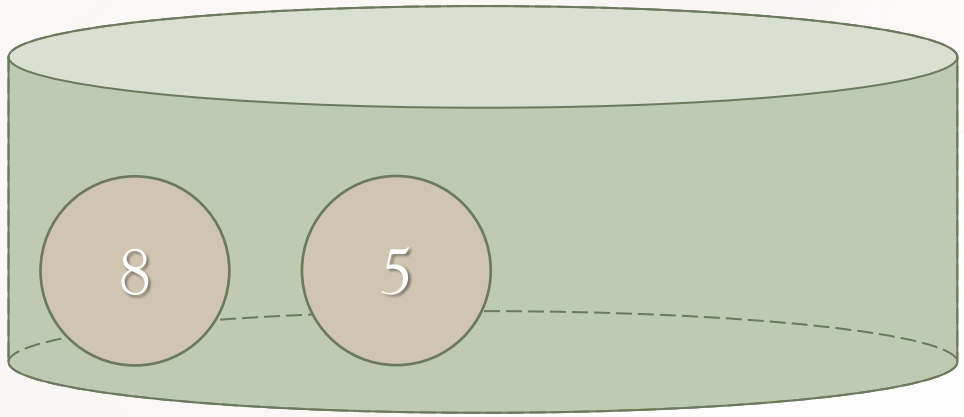
agregar (5)
conjuntoVacio () = false
pertenece (2) = true
pertenece (5) = true

Conjuntos - Especificación



elegir () = 2
conjuntoVacio () = false
pertenece (2) = true
pertenece (5) = true

Conjuntos - Especificación



sacar (2)
conjuntoVacio () = false
pertenece (2) = false
pertenece (5) = true

Conjuntos - Aclaraciones

- *Debido a que la estructura no tiene un orden, cuando recuperamos un dato la estructura nos devuelve uno* ***cualquiera*** *que pertenece a ella.*
- *Al agregar un elemento ya existente, como no se aceptan repetidos, no se hace nada.*
- *Cuando queremos* ***eliminar*** *un valor debemos indicarle cuál.*
- *Si se elimina un elemento que no existe, no se hace nada.*

Especificación - Interfaz

```
public interface ConjuntoTDA {
  void inicializarConjunto( );
  void agregar(int x); //conjunto inicializado
  void sacar(int x); //conjunto inicializado
  int elegir( ); //conjunto inicializado y no vacío
  boolean pertenece(int x); //conjunto inicializado
  boolean conjuntoVacio( ); //conjunto
        inicializado
}
```

Uso - Ejemplos

- *Vamos a escribir un método que nos permita determinar si un conjunto Conjunto1 incluye a otro conjunto Conjunto2.*

Elegir

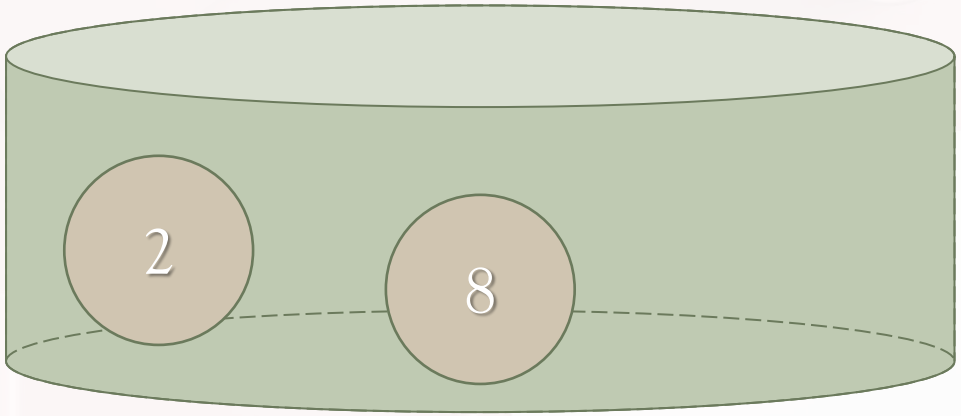
Implementación estática Estrategia 1

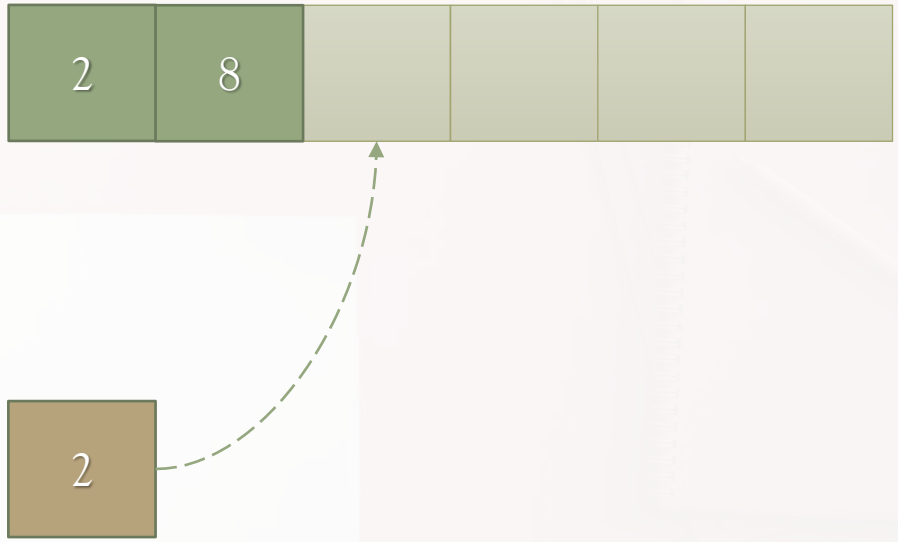
- *Se define un arreglo que contendrá los elementos del conjunto.*
- *Una variable entera indicará la cantidad de posiciones utilizadas en el arreglo.*
- *Como en un conjunto no puede haber repeticiones, antes de **agregar** un nuevo elemento debemos asegurarnos de que no esté ya en él. Como no importa el orden, lo colocamos en la primera posición disponible.*

Implementación estática Estrategia 1

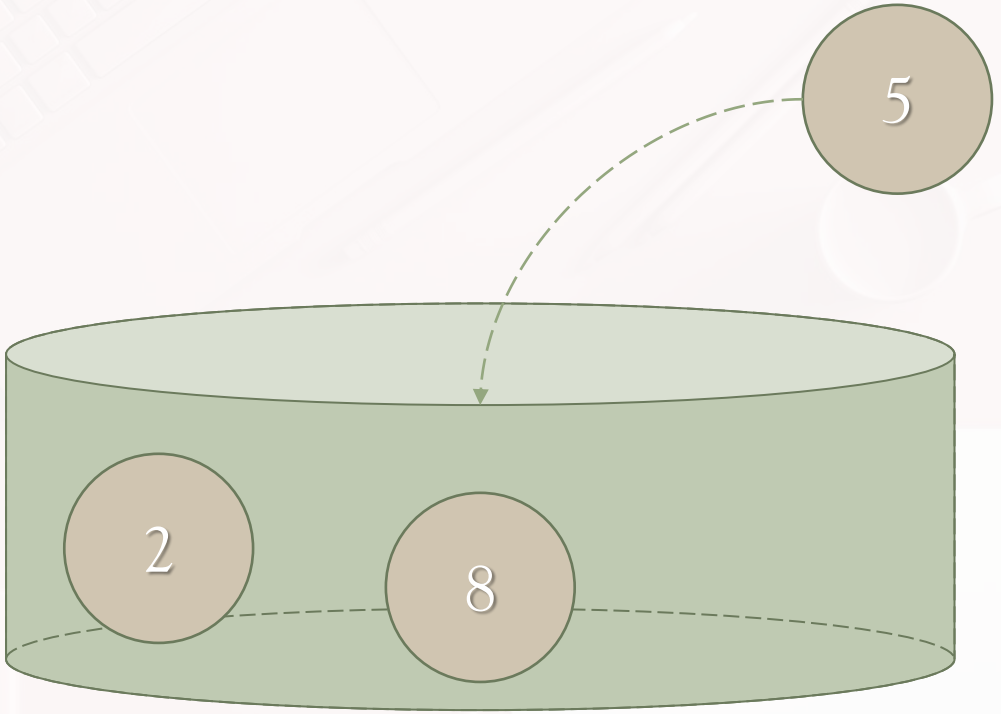
- *Para determinar si un elemento dado **pertenece** al conjunto es necesario recorrerlo enteramente.*
- *Para **eliminar** un elemento, se lo sobreescribe con el último elemento del arreglo (posición cant-1), evitando el desplazamiento de todos los elementos.*
- *Al **elegir** un valor vamos a recuperar cualquier elemento (arbitrariamente).*

Conjuntos - Implementación estática

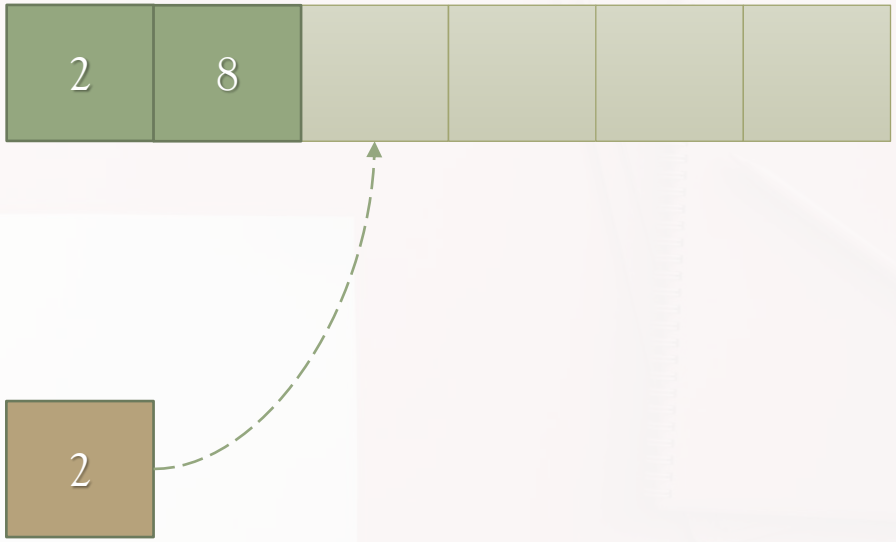




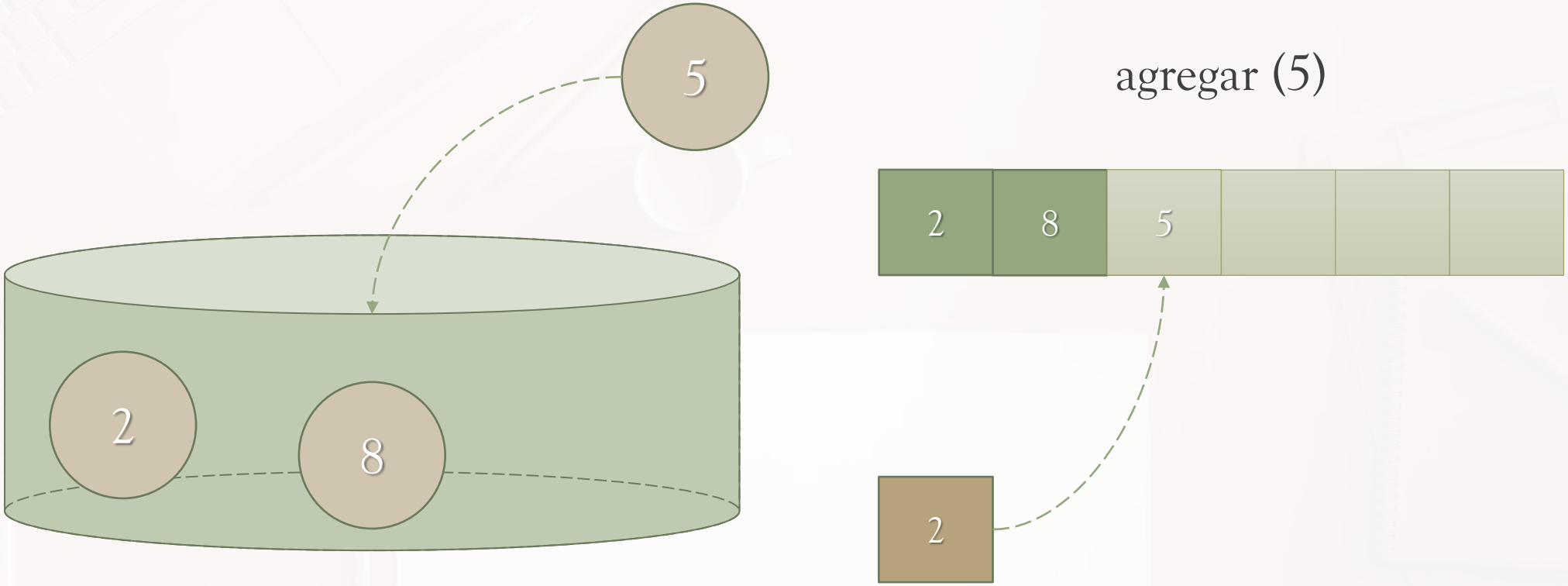
Conjuntos - Implementación estática



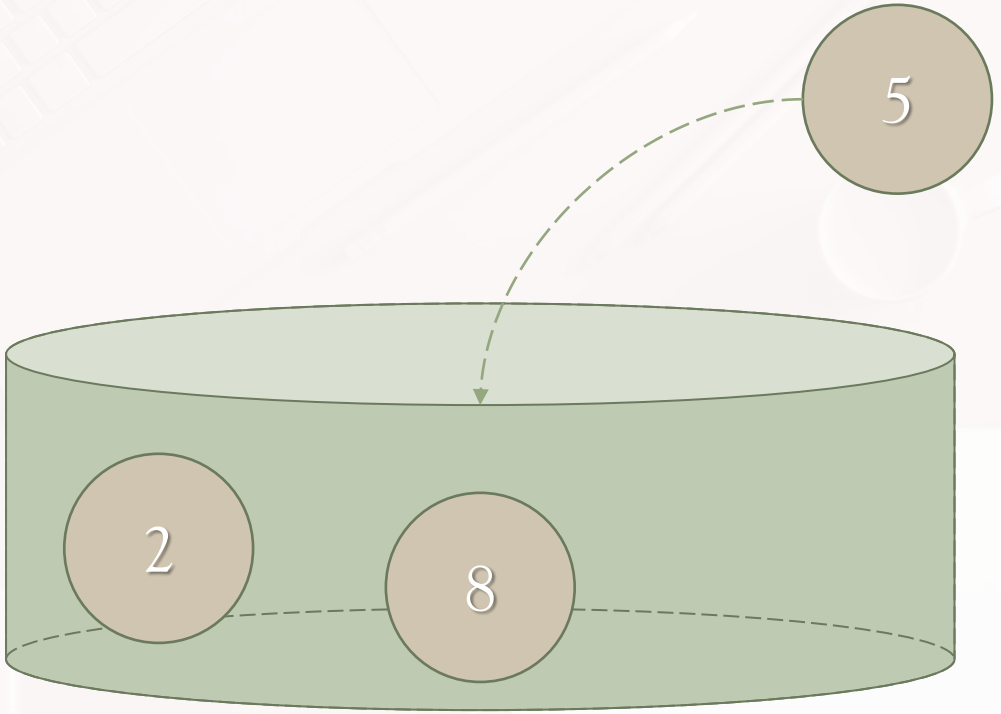
agregar (5)



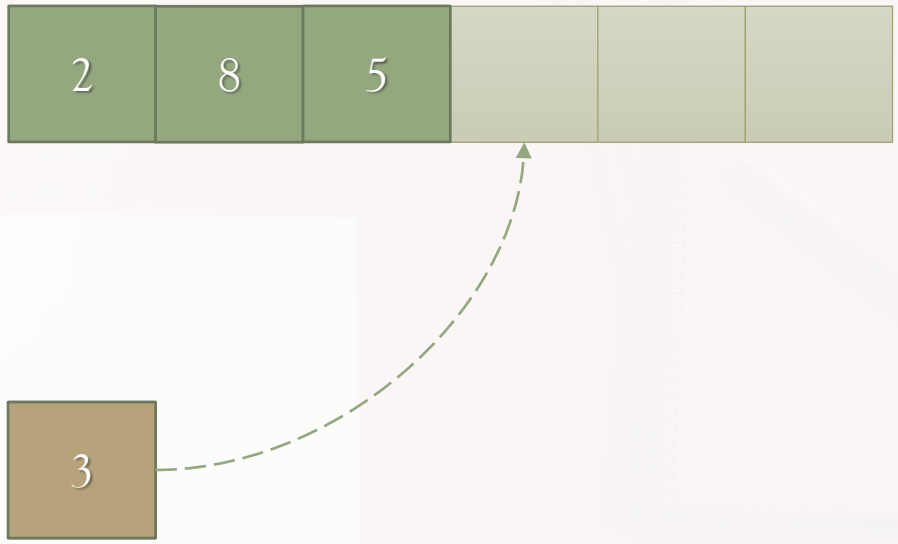
Conjuntos - Implementación estática



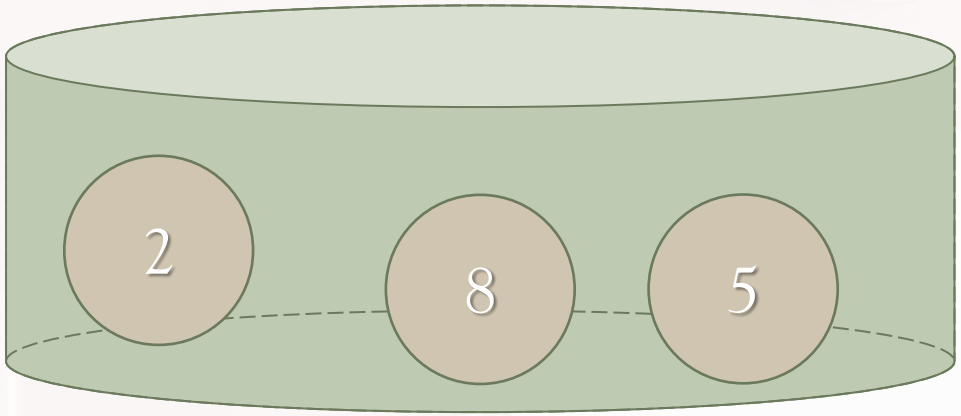
Conjuntos - Implementación estática

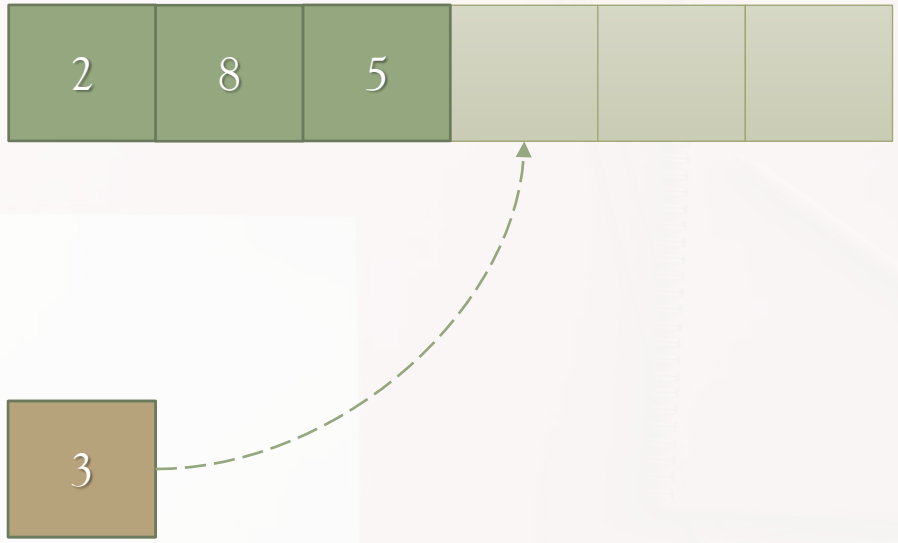


agregar (5)

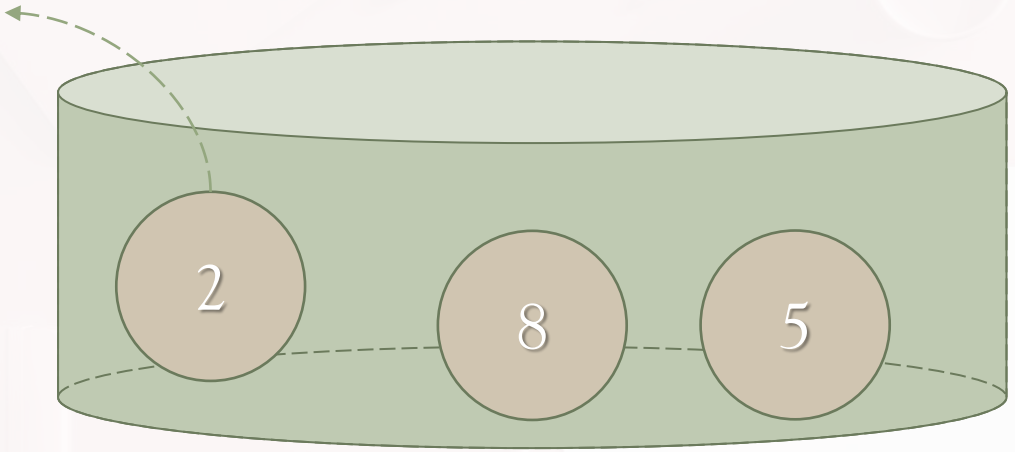


Conjuntos - Implementación estática

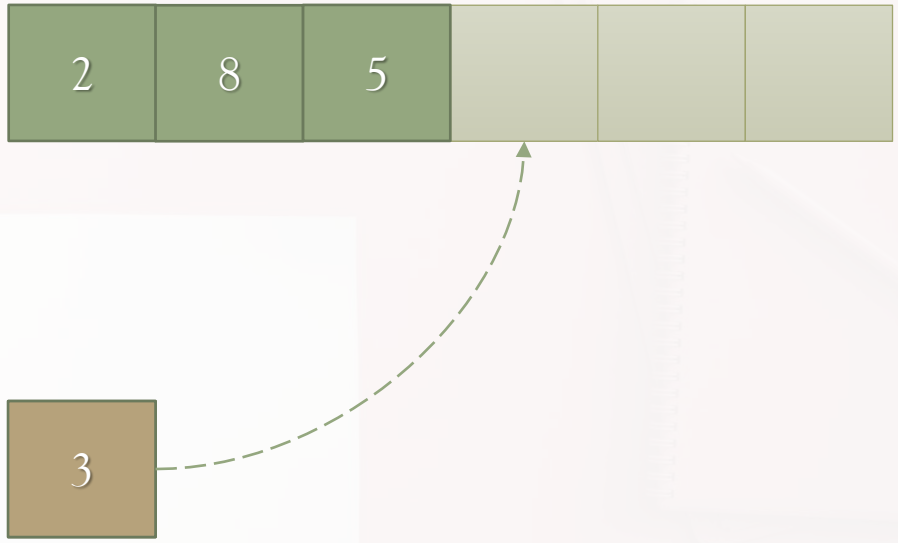




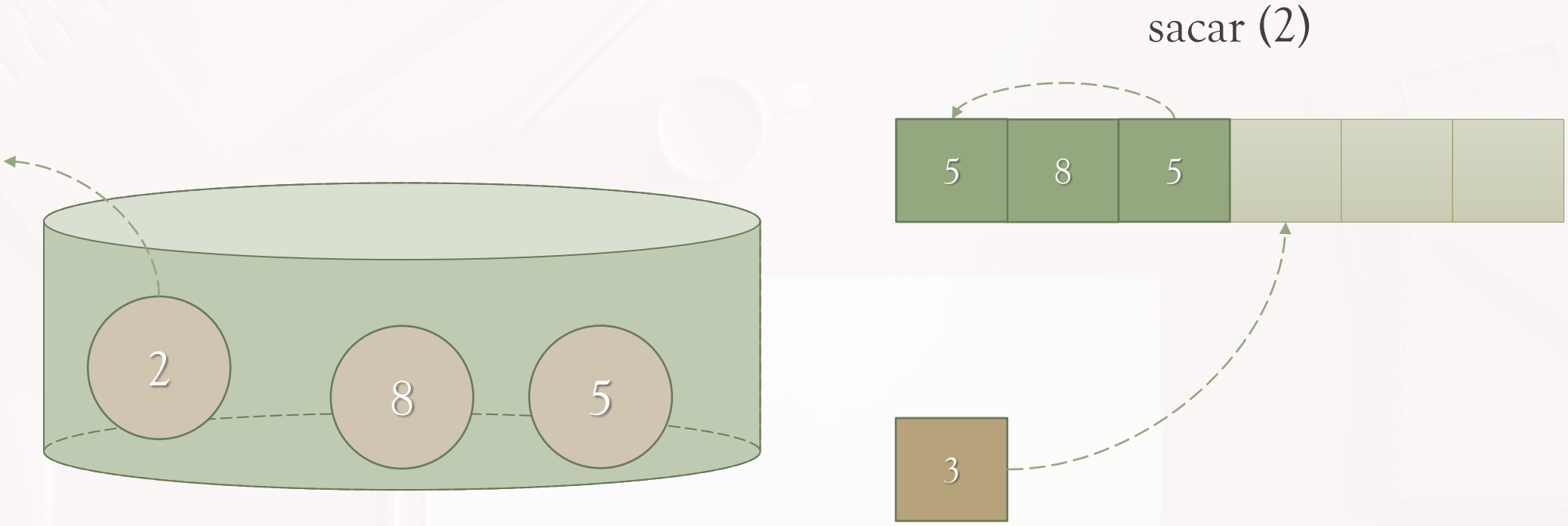
Conjuntos - Implementación estática



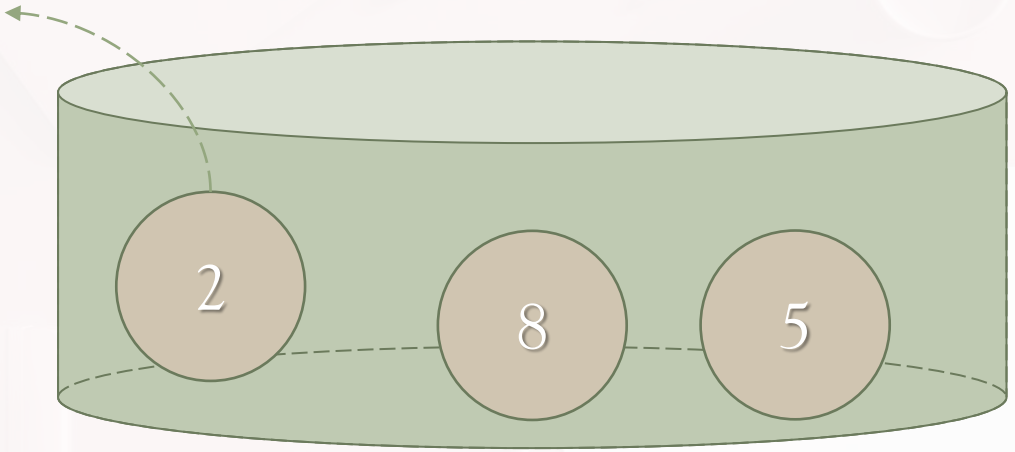
sacar (2)



Conjuntos - Implementación estática

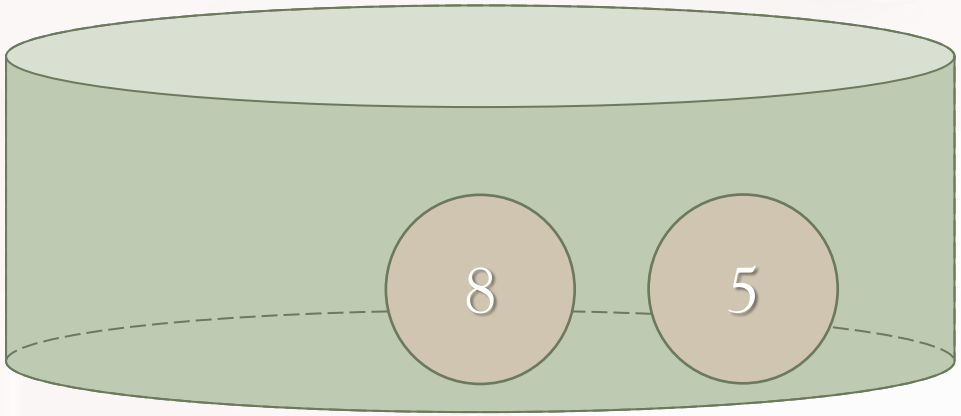


Conjuntos - Implementación estática



sacar (2)

Conjuntos - Implementación estática



Implementación estática Aclaraciones

- La ***eliminación*** de un elemento del vector arr se representa dejándolo afuera de la parte del arreglo delimitada por la variable indice; a los efectos prácticos, cualquier elemento arr[i] situado en una posición i ≥ indice ***no existe más en el conjunto***.
- Tanto el vector arr, como el entero índice ***no son accesibles desde afuera de la implementación*** (son privados).


¡Muchas Gracias!

Bibliografía

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