Programación II
Diccionarios Múltiples

2C 2023 TN – Ing. Elizabeth Barrera

Temas

- *TDA*
- *Diccionarios múltiples*
- *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.


Diccionarios
Múltiples

Diccionarios Múltiples

Un diccionario múltiple es una estructura de datos para almacenar un grupo de elementos.

Un diccionario ***tiene un conjunto de claves*** *y cada clave tiene* ***asociado un conjunto de valores****.*

Cuando se le presenta una clave, el diccionario devuelve el conjunto de valores asociado.

Especificación

*Las operaciones que necesitamos son agregar un elemento con su clave (que llamaremos **agregar**), eliminar un elemento a través de la eliminación de su clave (que llamaremos **eliminar**), eliminar un elemento de la colección de valores de una clave (que llamaremos **eliminarValor**), recuperar el conjunto de valores correspondiente a una clave (que llamaremos **recuperar**) y obtener el conjunto de claves (que llamaremos **claves**). Necesitaremos, como siempre, una operación de inicializar un diccionario (que llamaremos **inicializarDiccionario**).*

Especificación - Operaciones

- ***inicializarDiccionario***: *permite inicializar la estructura del diccionario.*
- ***agregar***: *dada una clave y un valor, agrega al diccionario el valor quedando asociado a la clave (se supone que el diccionario está inicializado).*
- ***eliminar***: *dada una clave elimina todos los valores asociados a la clave, y por consiguiente la clave (se supone que el diccionario está inicializado).*

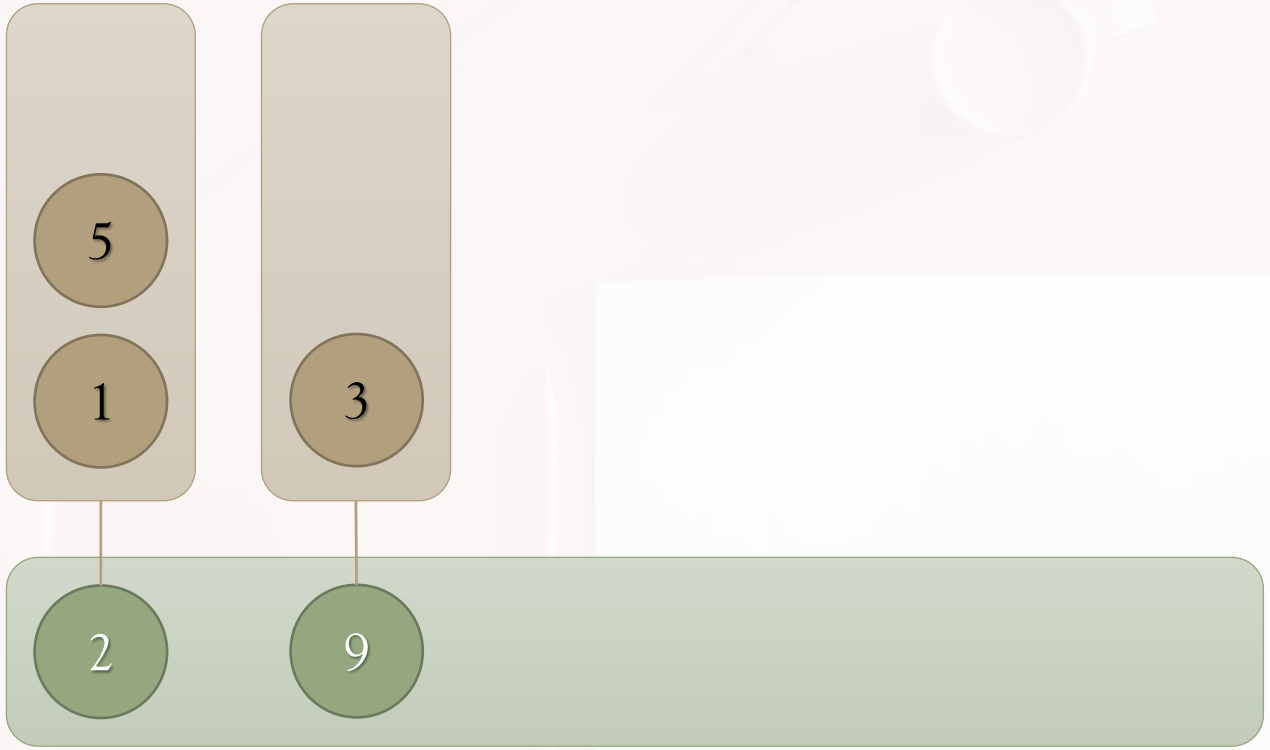
Recordar que:
*Las **precondiciones**, son condiciones que deben cumplirse antes de la ejecución de la operación.*

Especificación - Operaciones

- ***eliminarValor***: *dada una clave y un valor se elimina el valor asociado a la clave (se supone que el diccionario está inicializado).*
- ***recuperar***: *dada una clave devuelve el conjunto de valores asociados a la misma (se supone que el diccionario está inicializado).*
- ***claves***: *devuelve el conjunto de todas las claves definidas en el diccionario (se supone que el diccionario está inicializado).*

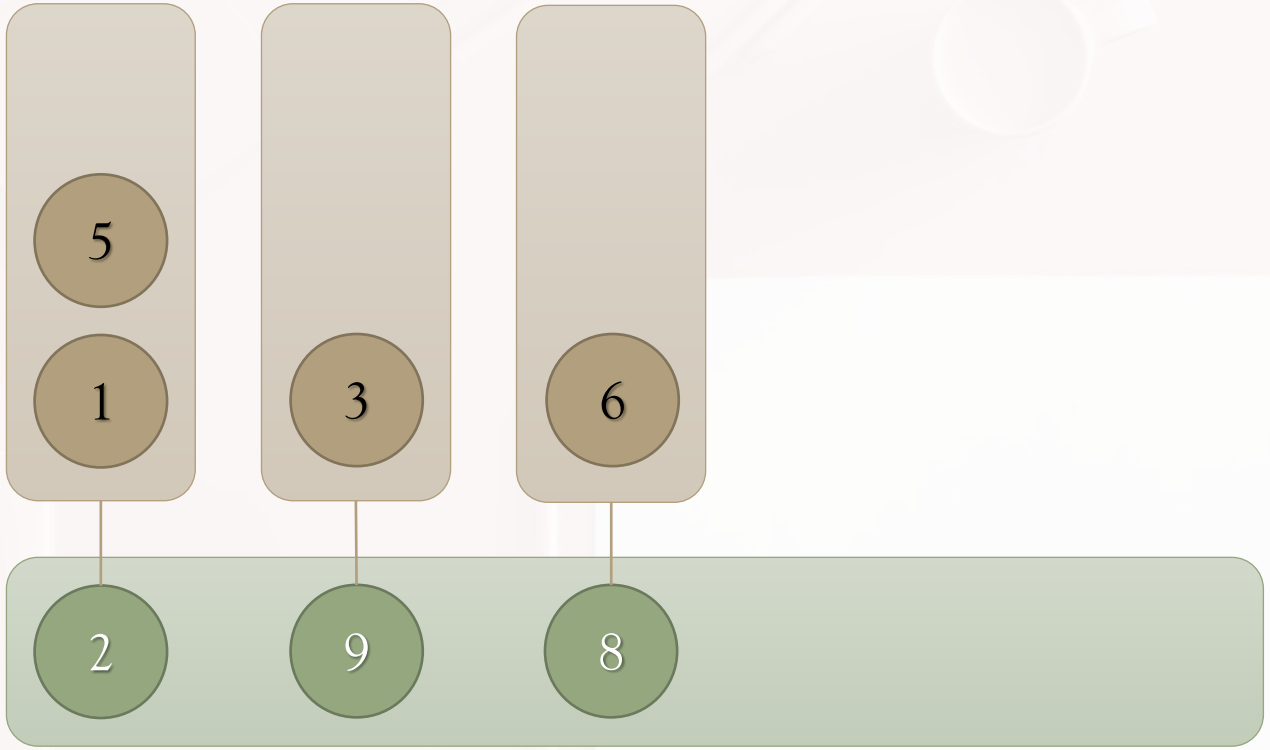
Recordar que:
*Las **precondiciones**, son condiciones que deben cumplirse antes de la ejecución de la operación.*

Diccionarios Múltiples Especificación



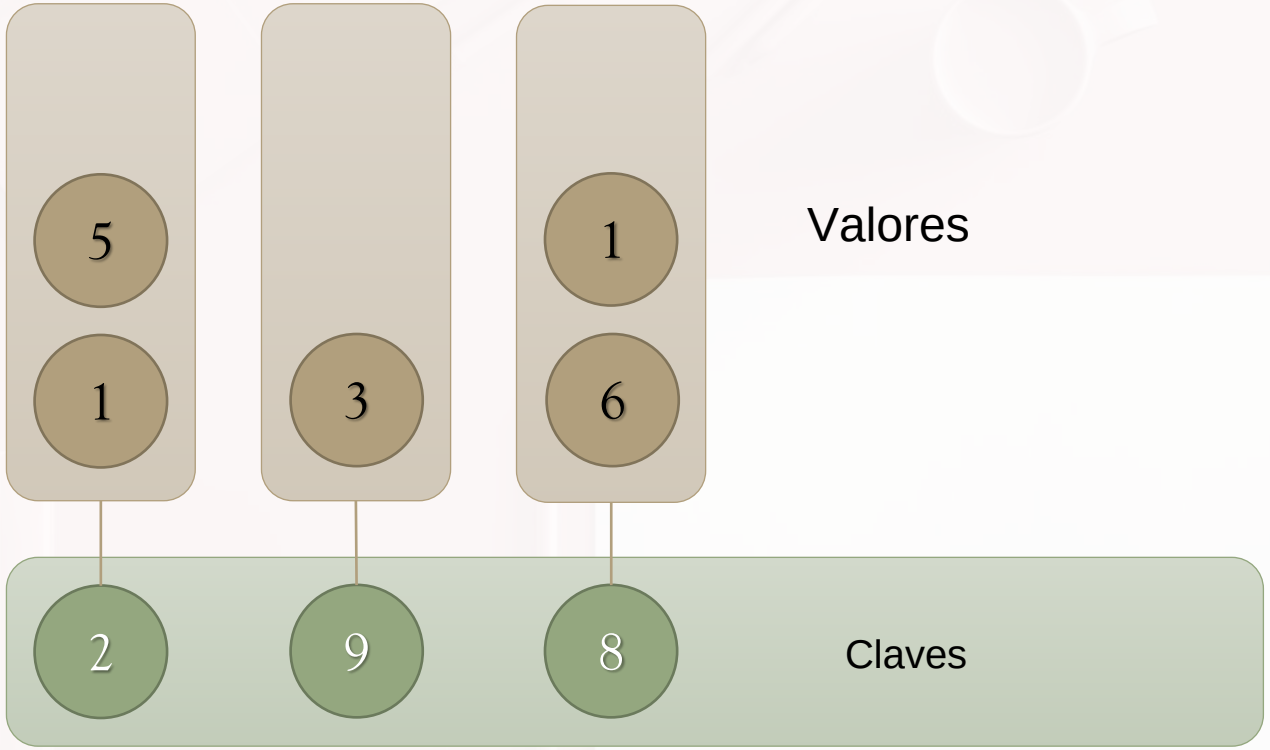
claves () = {2, 9}

Diccionarios Múltiples Especificación



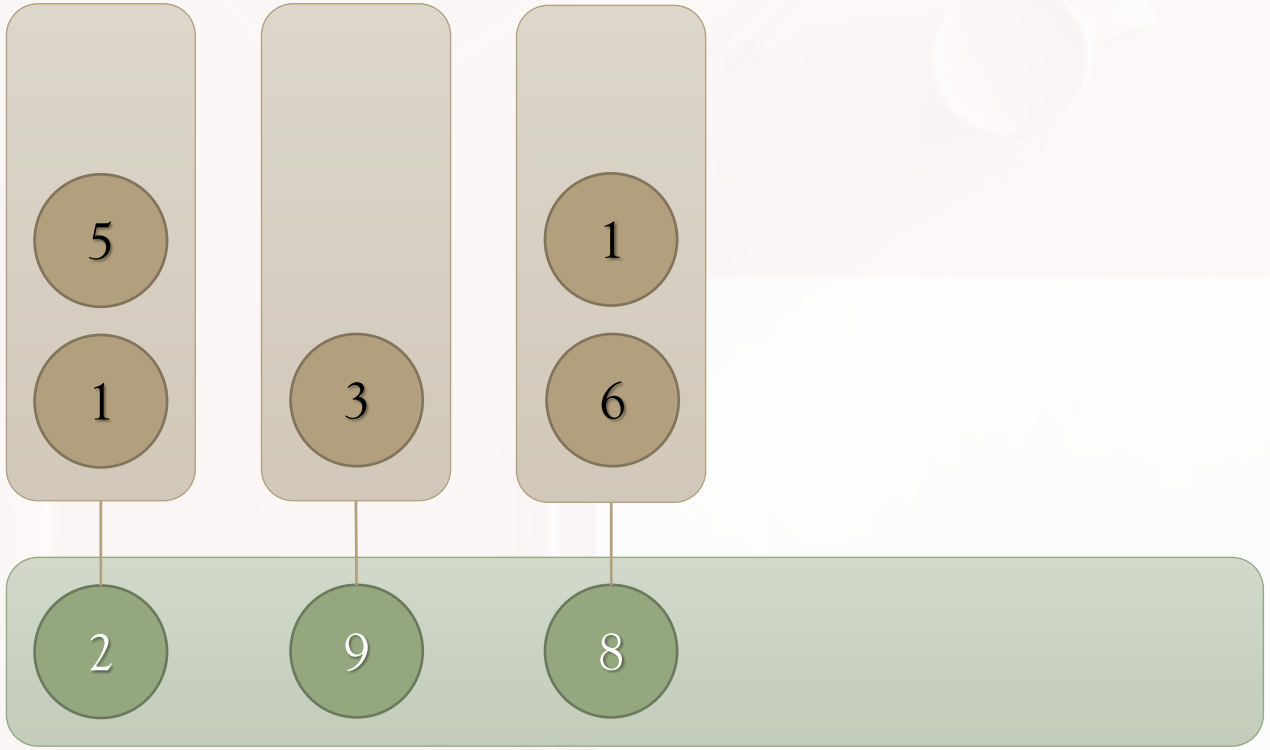
agregar (8, 6)
claves () = {2, 9, 8}

Diccionarios Múltiples Especificación



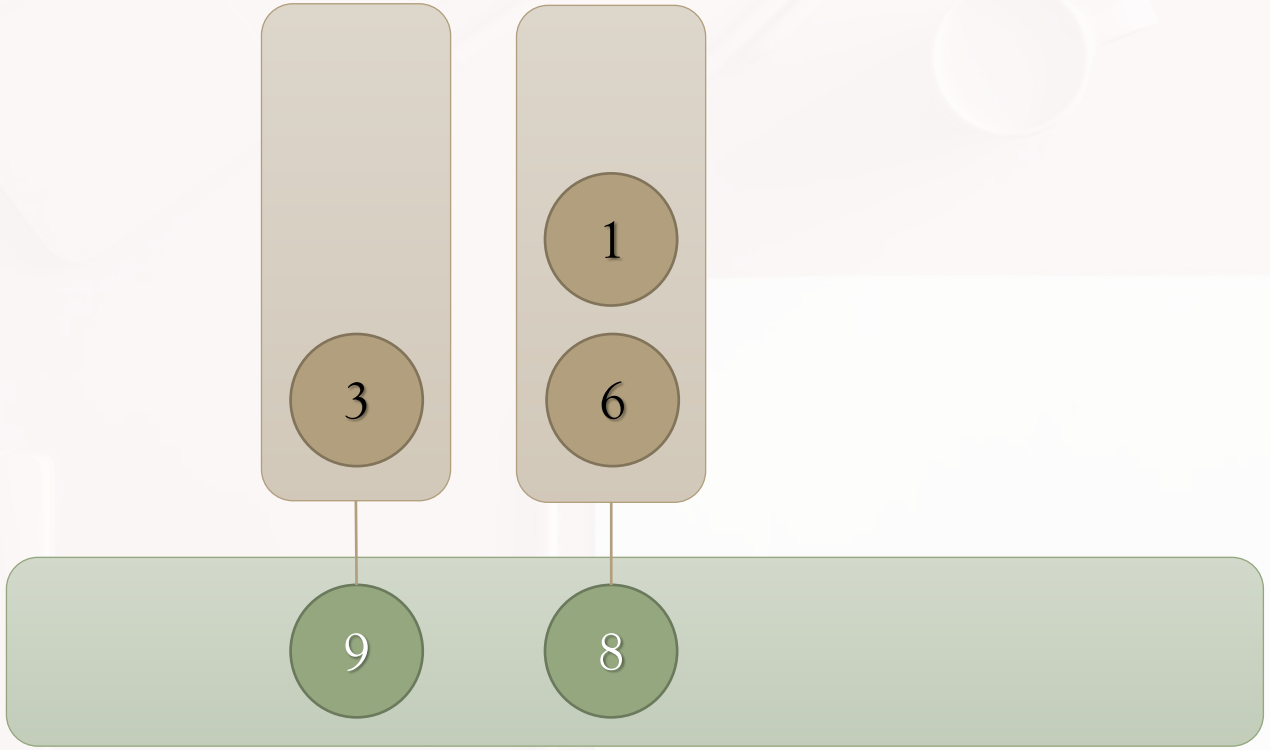
agregar (8, 1)
claves () = {2, 9, 8}

Diccionarios Múltiples Especificación



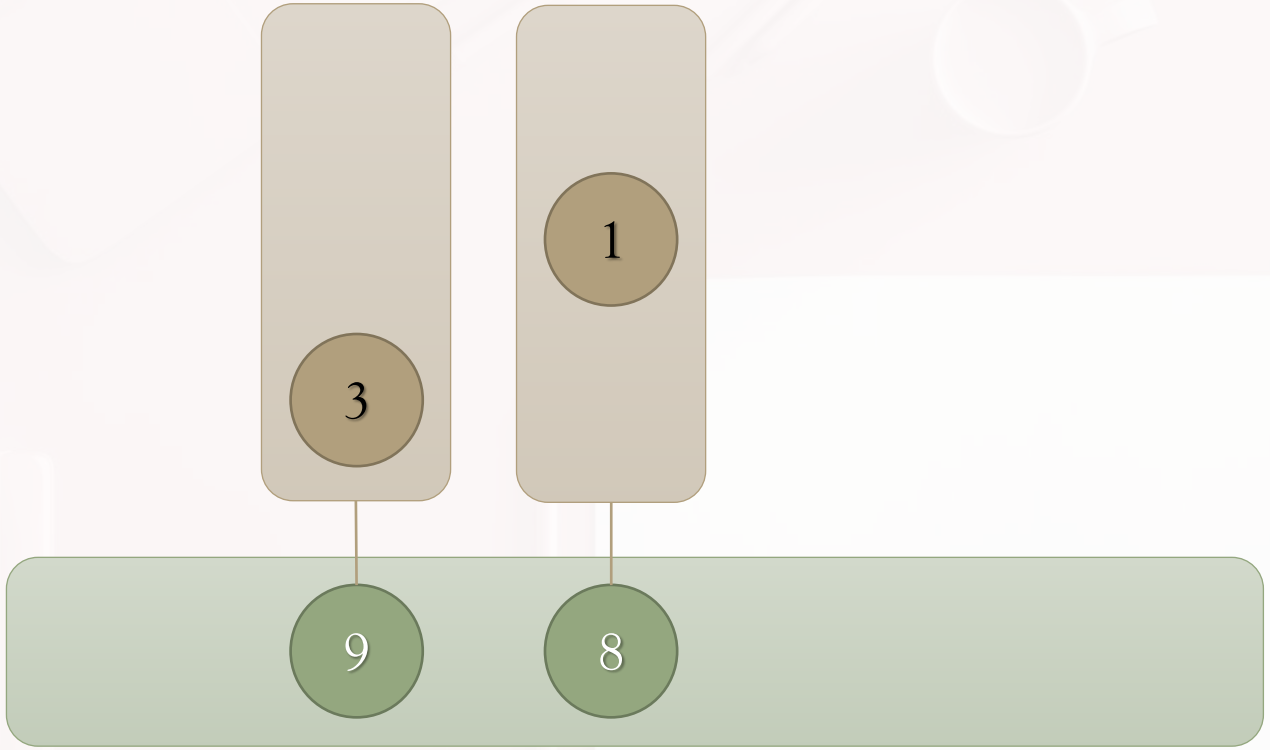
recuperar (2) = {1, 5}
claves () = {2, 9, 8}

Diccionarios Múltiples Especificación



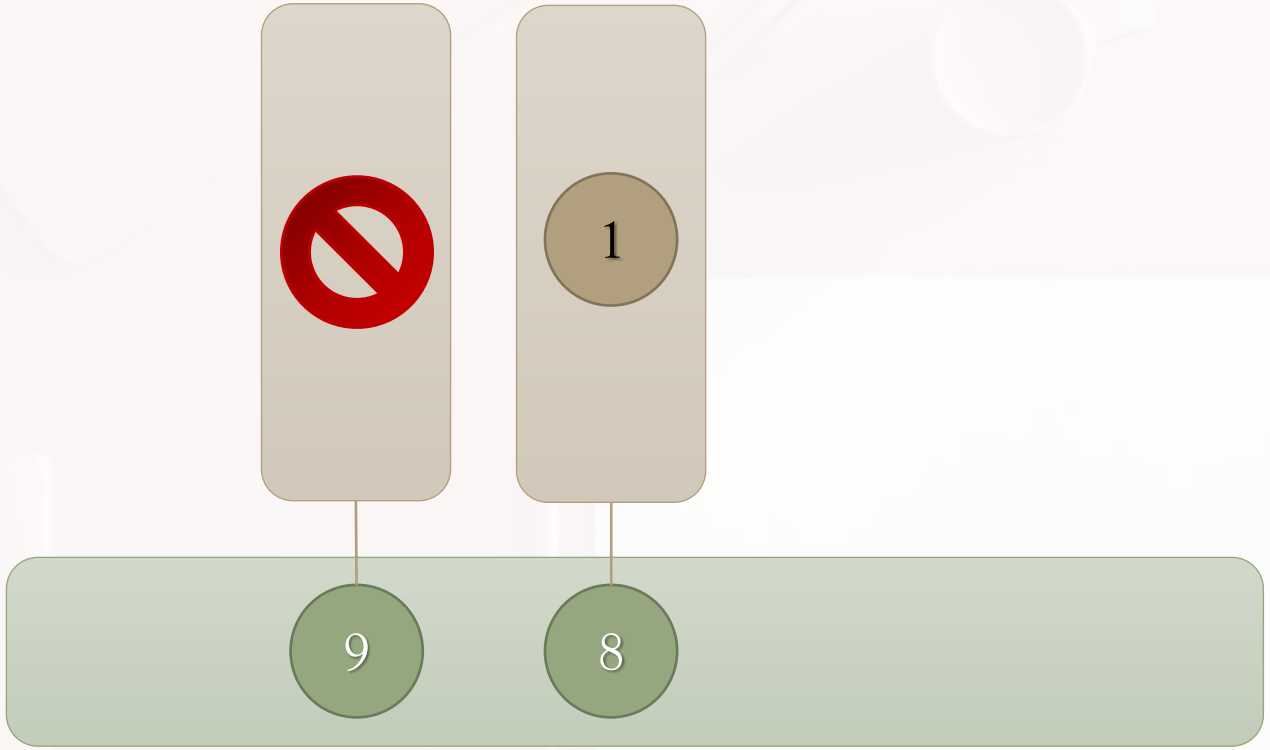
eliminar (2)
claves () = {9, 8}

Diccionarios Múltiples Especificación



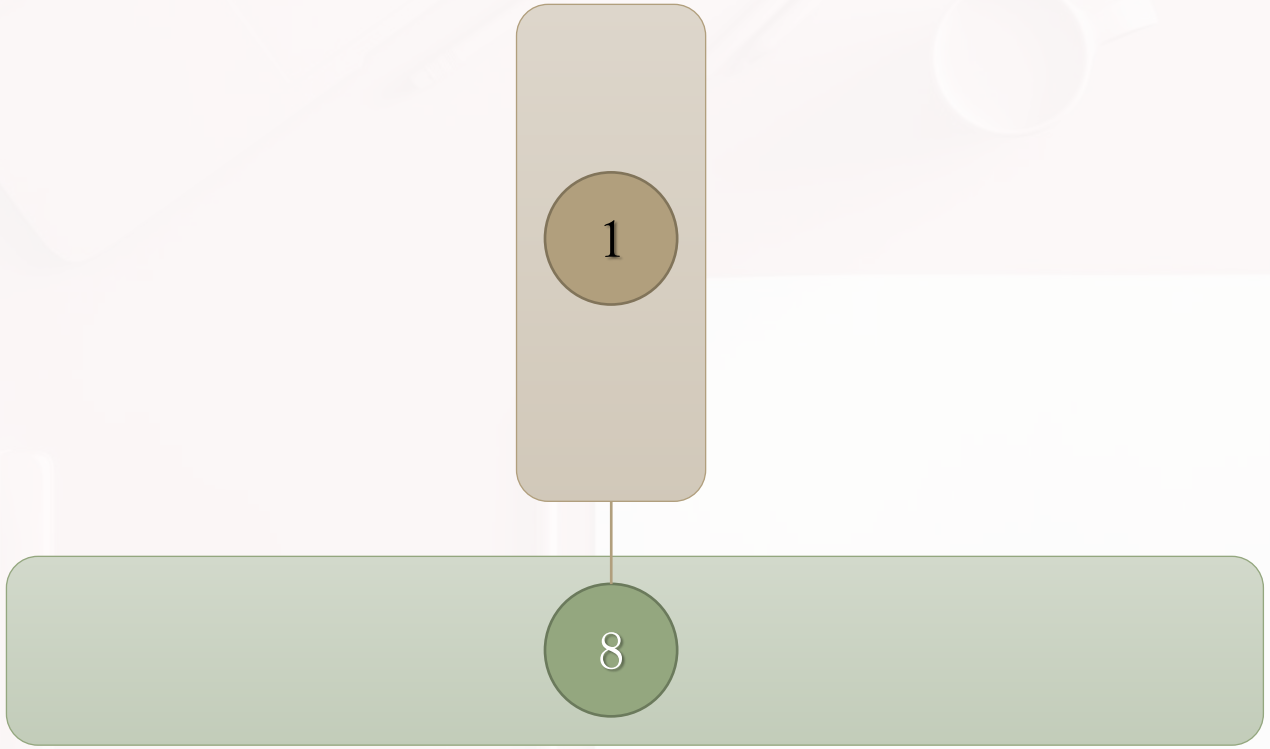
eliminarValor (8, 6)
claves () = {9, 8}

Diccionarios Múltiples Especificación



eliminarValor (9, 3)
claves () = {8}

Diccionarios Múltiples Especificación



eliminarValor (9, 3)
claves () = {8}

Aclaraciones

- *Una misma clave puede tener asociada un conjunto de valores, pero esos **valores no se pueden repetir**.*
- *El diccionario no puede contener claves sin valores asociados.*
- *Al eliminar o eliminarValor si la clave o el valor no existen, no se hace nada.*
- *Al eliminarValor si la clave no tiene otros valores asociados se debe eliminar la misma.*
- *Al recuperar si la clave dada no pertenece al diccionario, se debe devolver un conjunto vacío.*

Especificación - Interfaz

```
public interface DiccionarioMultipleTDA {
  void inicializarDiccionario( );
  void agregar(int clave, int valor); //diccionario
       inicializado
  void eliminar(int clave); //diccionario inicializado
  void eliminarValor(int clave, int valor);
       //diccionario inicializado
  ConjuntoTDA recuperar(int clave); //diccionario
       inicializado
  ConjuntoTDA claves( ); //diccionario inicializado
}
```

Uso

Uso - Ejemplos

- *Vamos a escribir un método que nos permita pasar los elementos de un diccionario simple DicSim a uno múltiple DicMul.*

Implementación

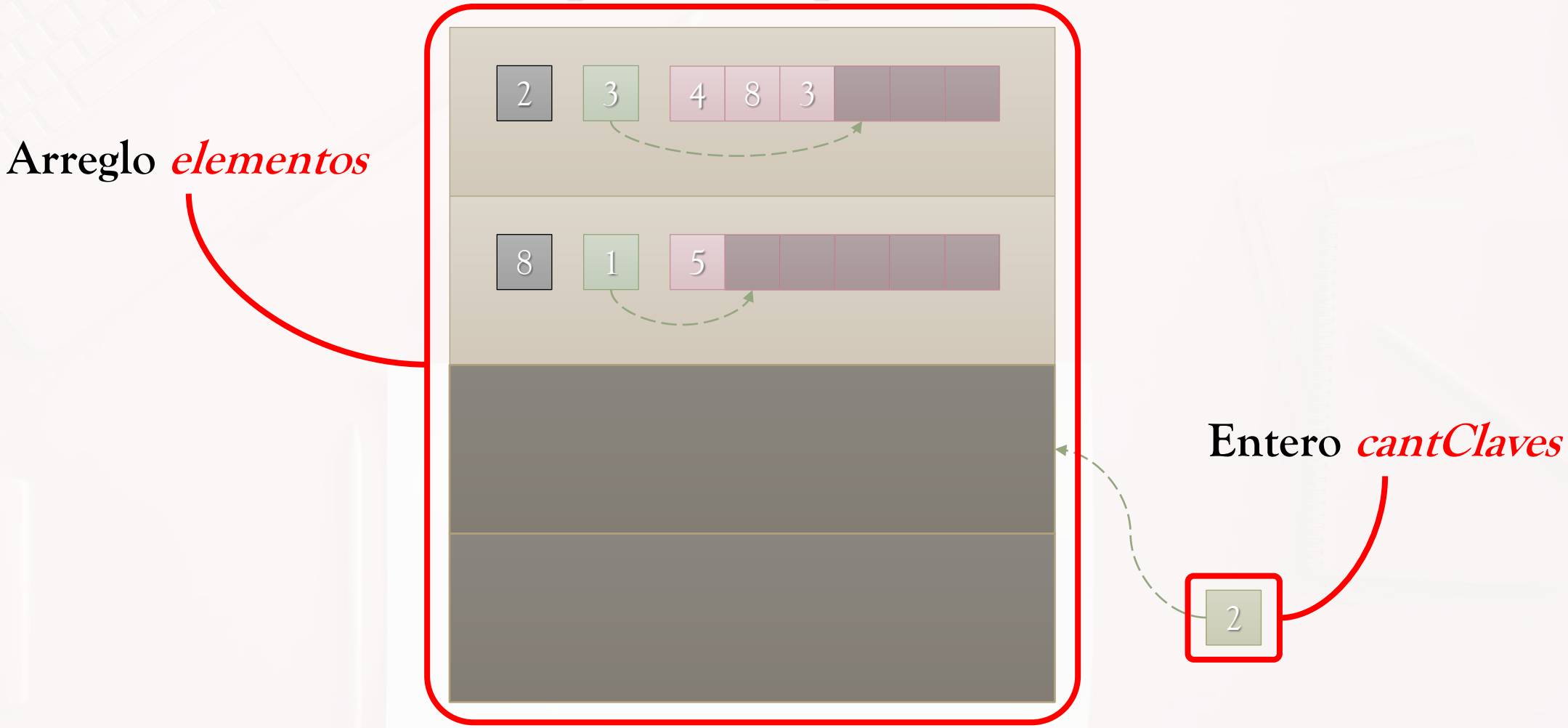
Implementación estática Estrategia 1

- Vamos a definir una estructura que contendrá una clave entera, un arreglo de enteros (los valores) y una variable entera que nos dará la cantidad de valores de la colección de la clave.
- El diccionario múltiple se representará como un arreglo de dicha estructura.
- Se utilizará una variable entera para indicar la cantidad de claves que hay en el arreglo de elementos.

Implementación estática Estrategia 1

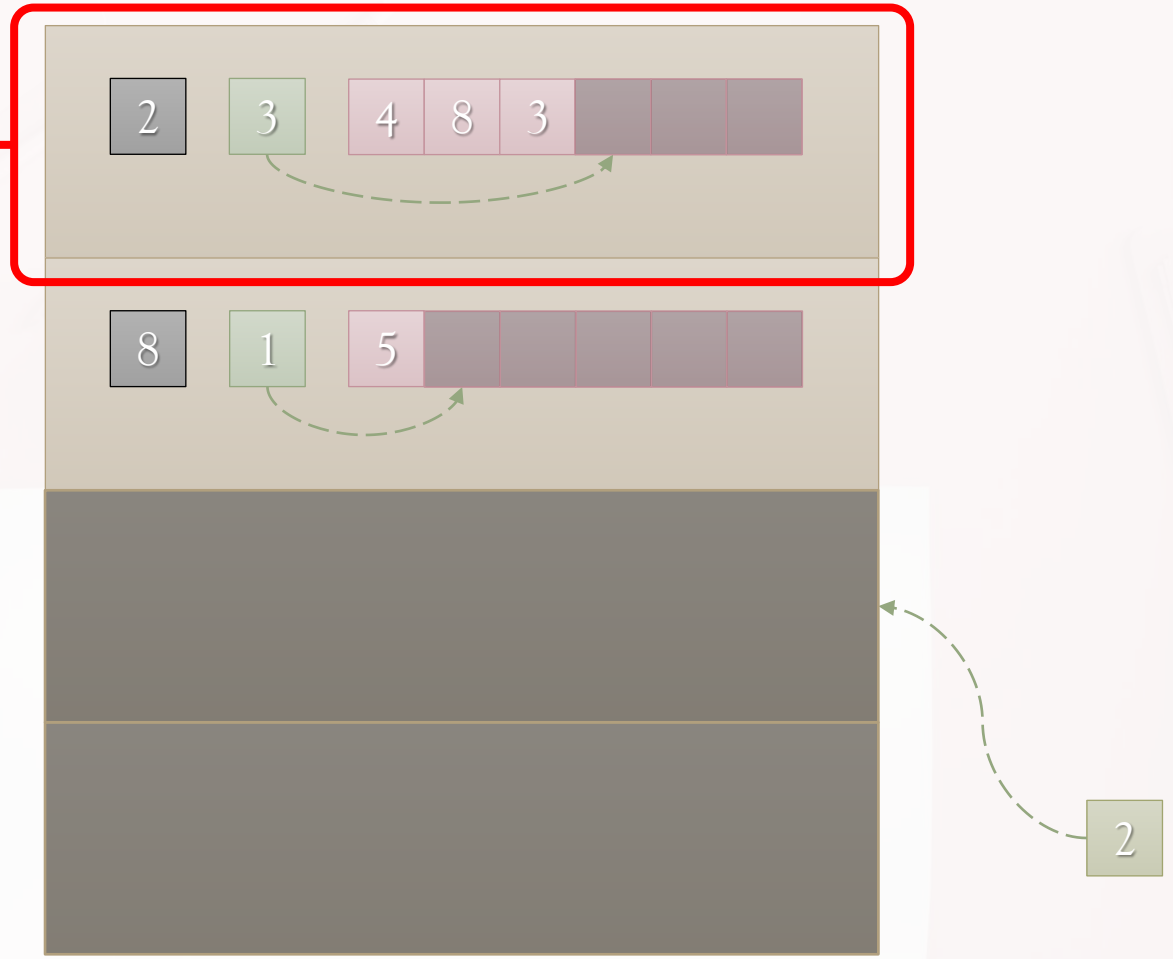
- *Se definirán dos métodos privados (para uso exclusivo de la implementación):* ***clave2Indice*** *que, dada una clave, devuelve el índice correspondiente en el arreglo de elementos; y* ***valor2Indice*** *que, dado un valor, devuelve el índice correspondiente en el arreglo de los valores.*
- *Cuando se* ***elimina*** *una clave (o un valor), se reemplaza el elemento eliminado con el de la última posición.*

Diccionarios Múltiples - Implementación estática



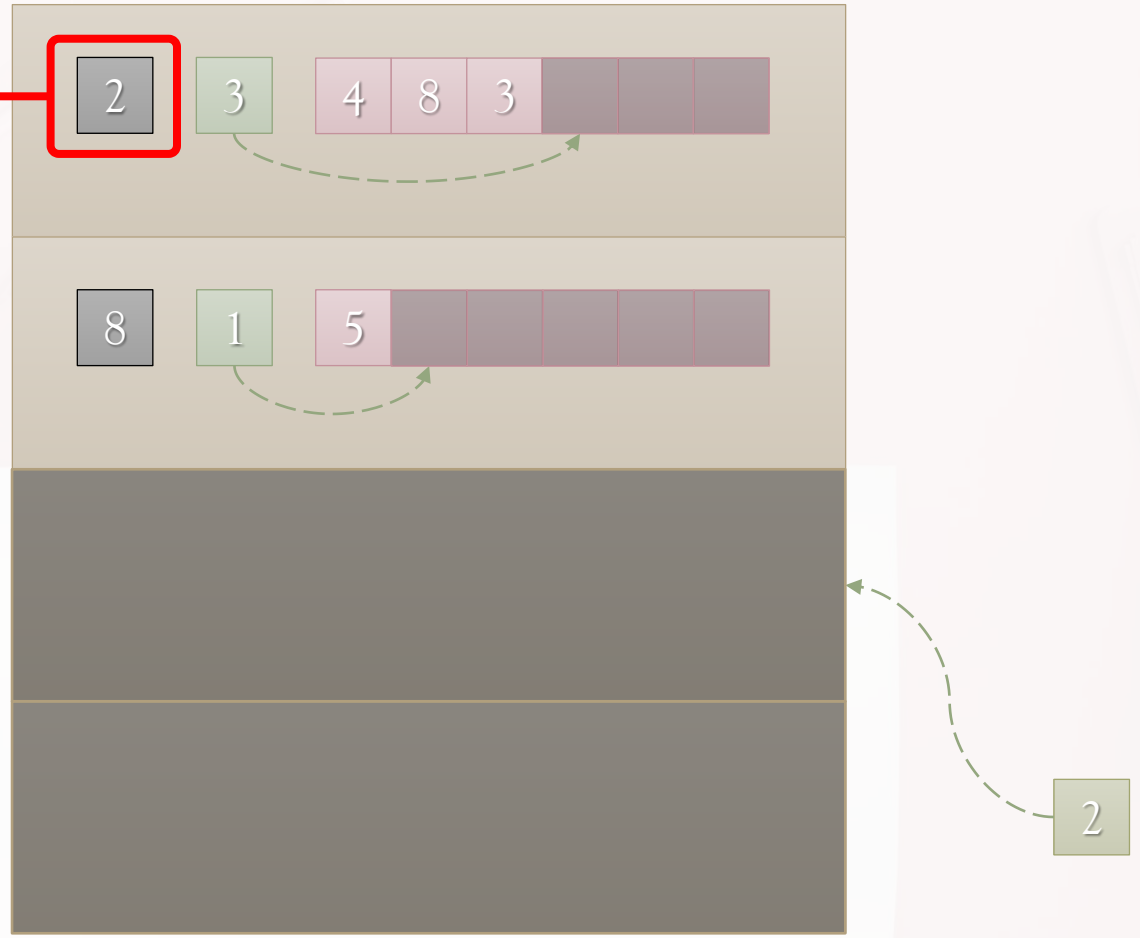
Diccionarios Múltiples - Implementación estática

Objeto *Elemento*



Diccionarios Múltiples - Implementación estática

[Elemento].*clave*



Diccionarios Múltiples - Implementación estática

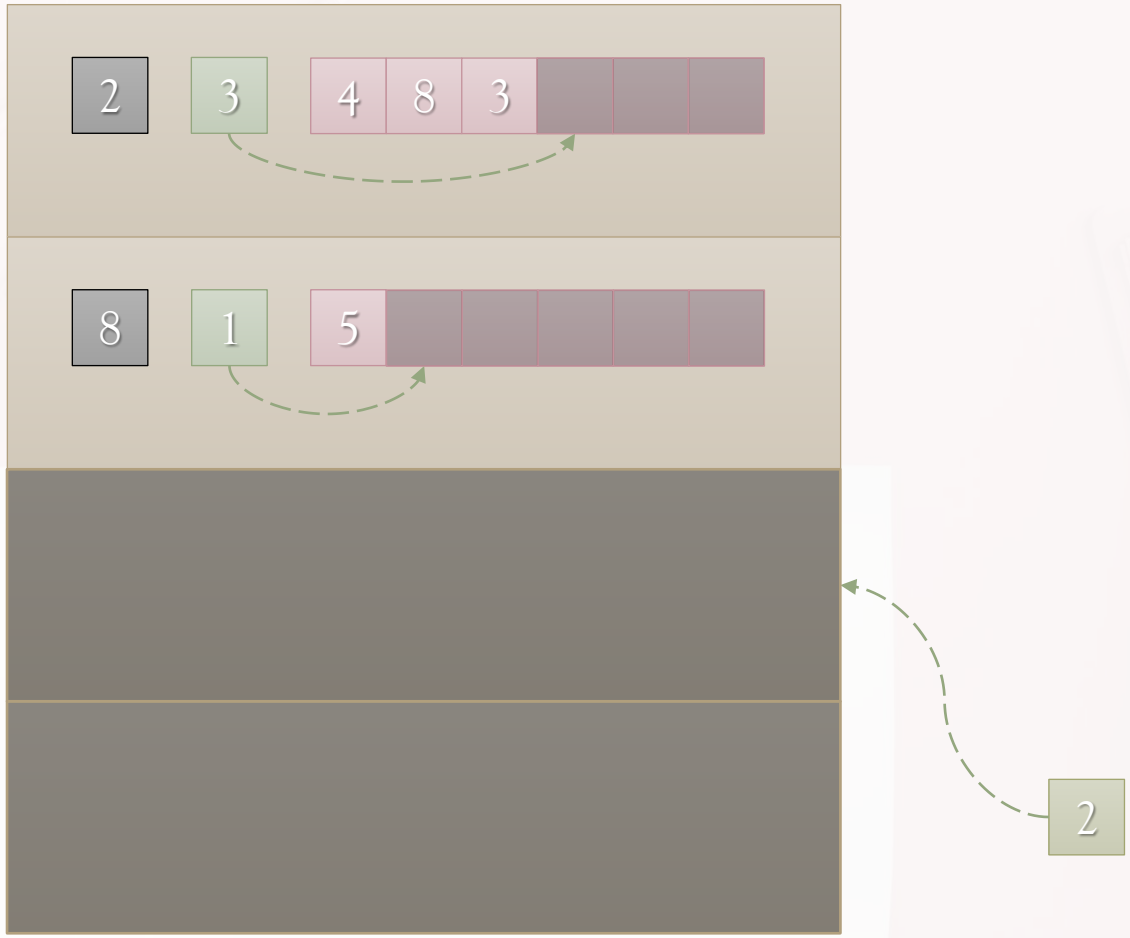
[Elemento].*cantValores*

Diccionarios Múltiples - Implementación estática

[Elemento].*valores*

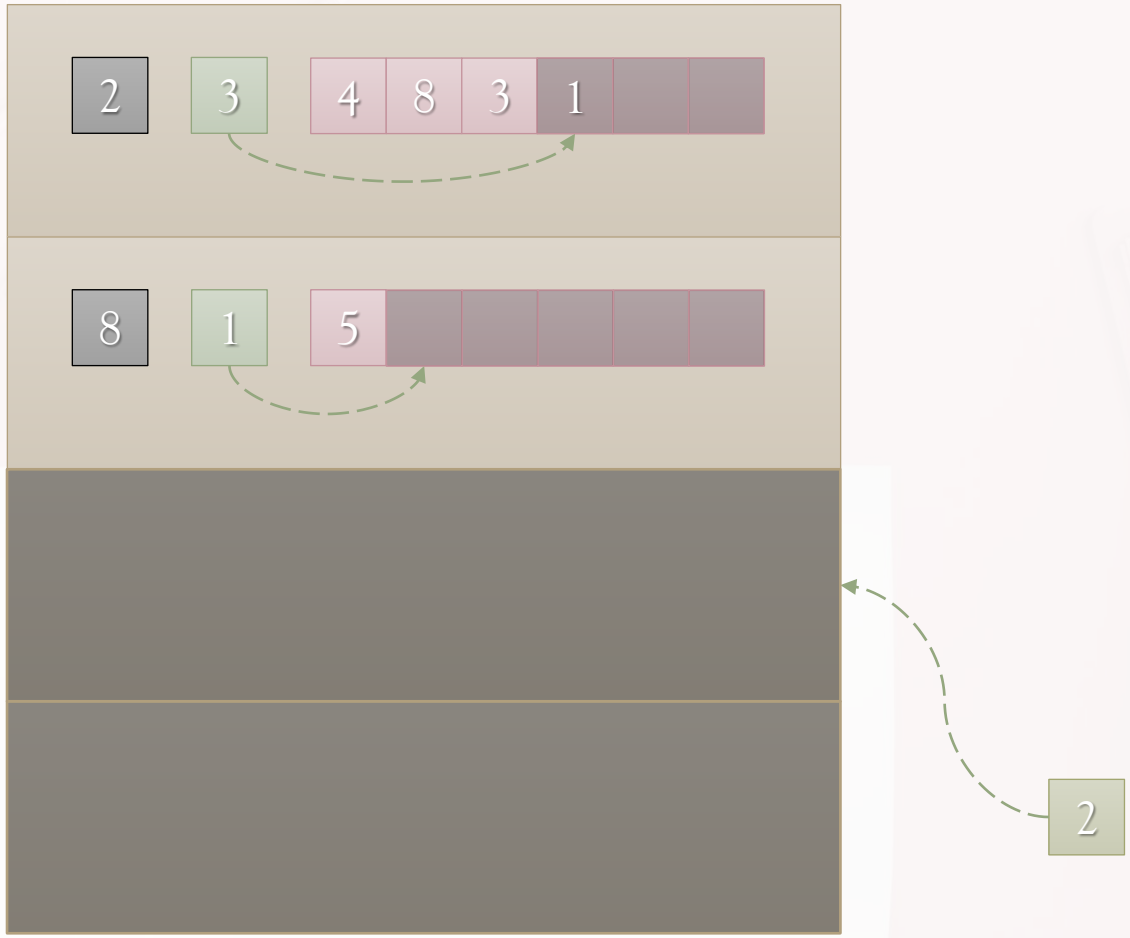
Diccionarios Múltiples - Implementación estática

agregar (2, 1)



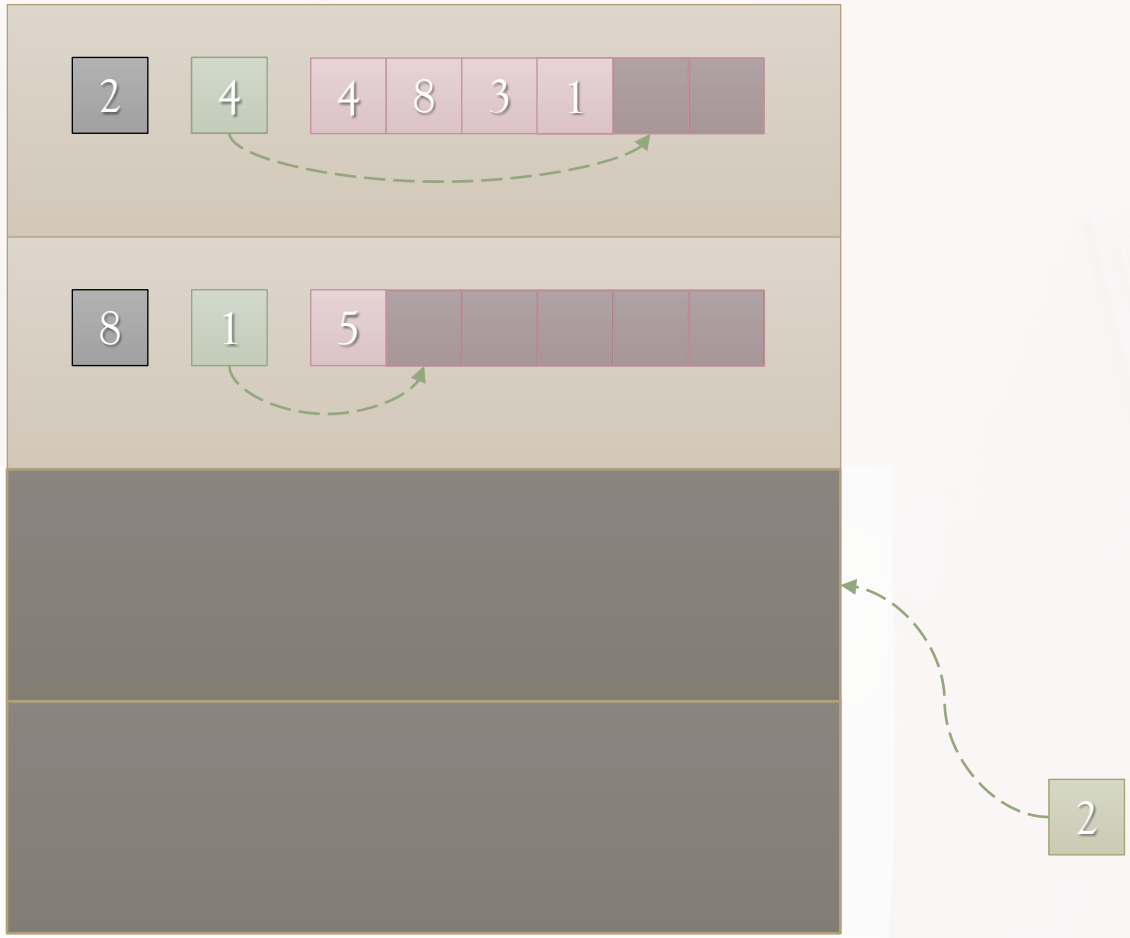
Diccionarios Múltiples - Implementación estática

agregar (2, 1)



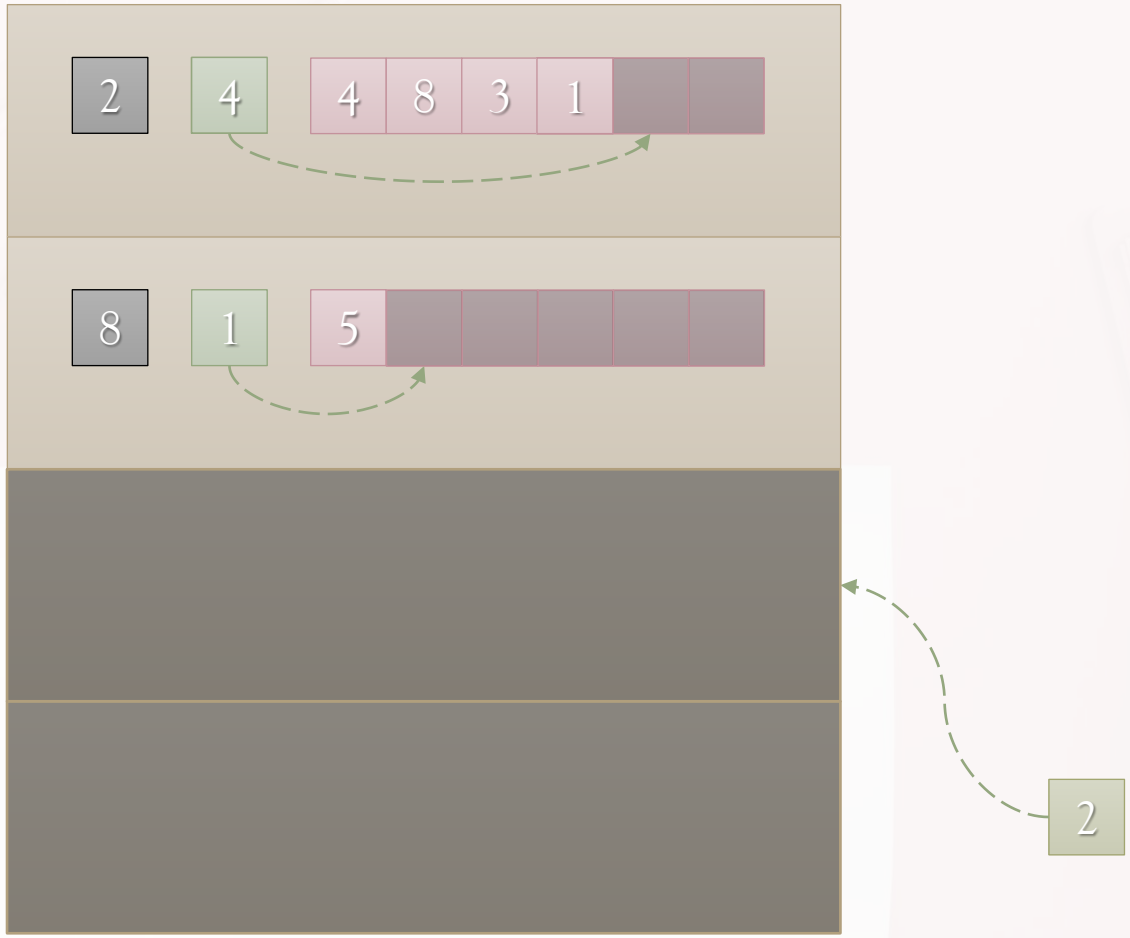
Diccionarios Múltiples - Implementación estática

agregar (2, 1)



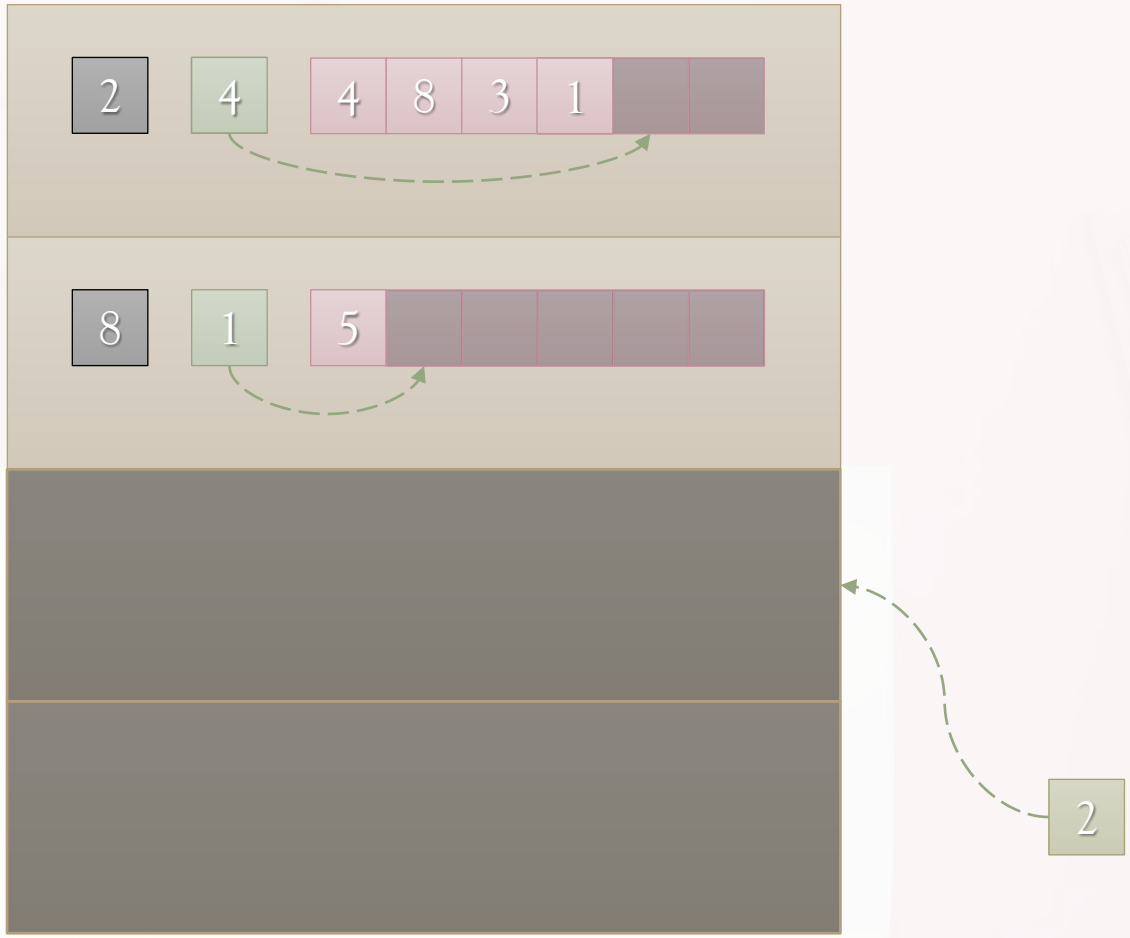
Diccionarios Múltiples - Implementación estática

agregar (8, 5)



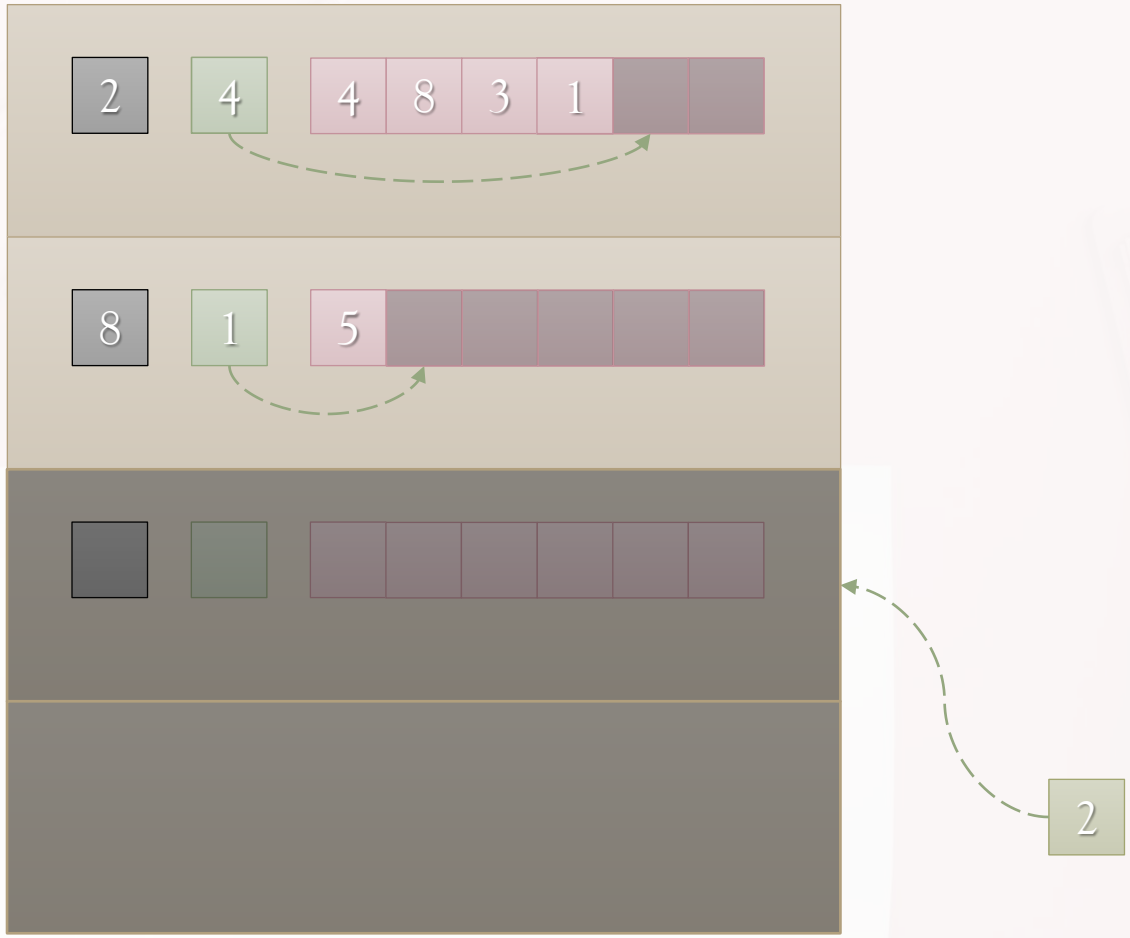
Diccionarios Múltiples - Implementación estática

agregar (3, 2)



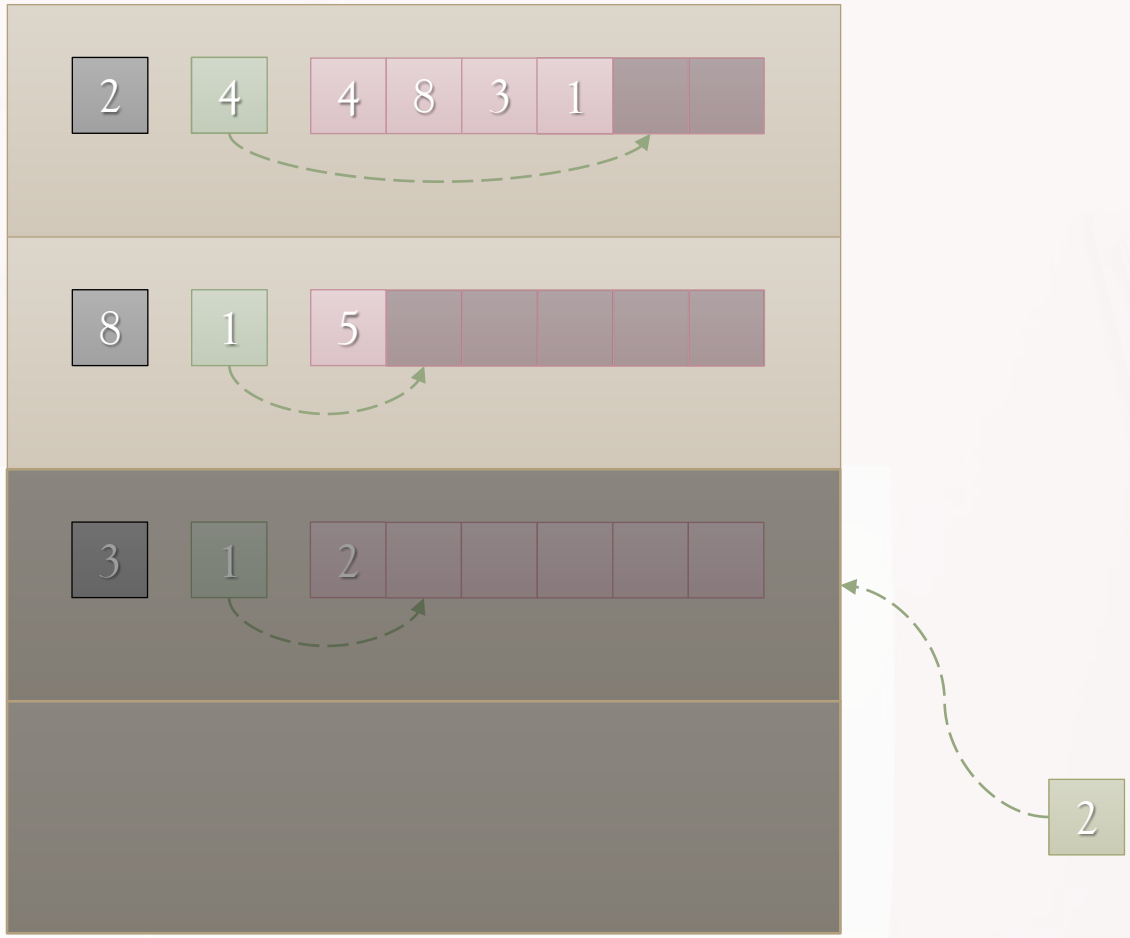
Diccionarios Múltiples - Implementación estática

agregar (3, 2)



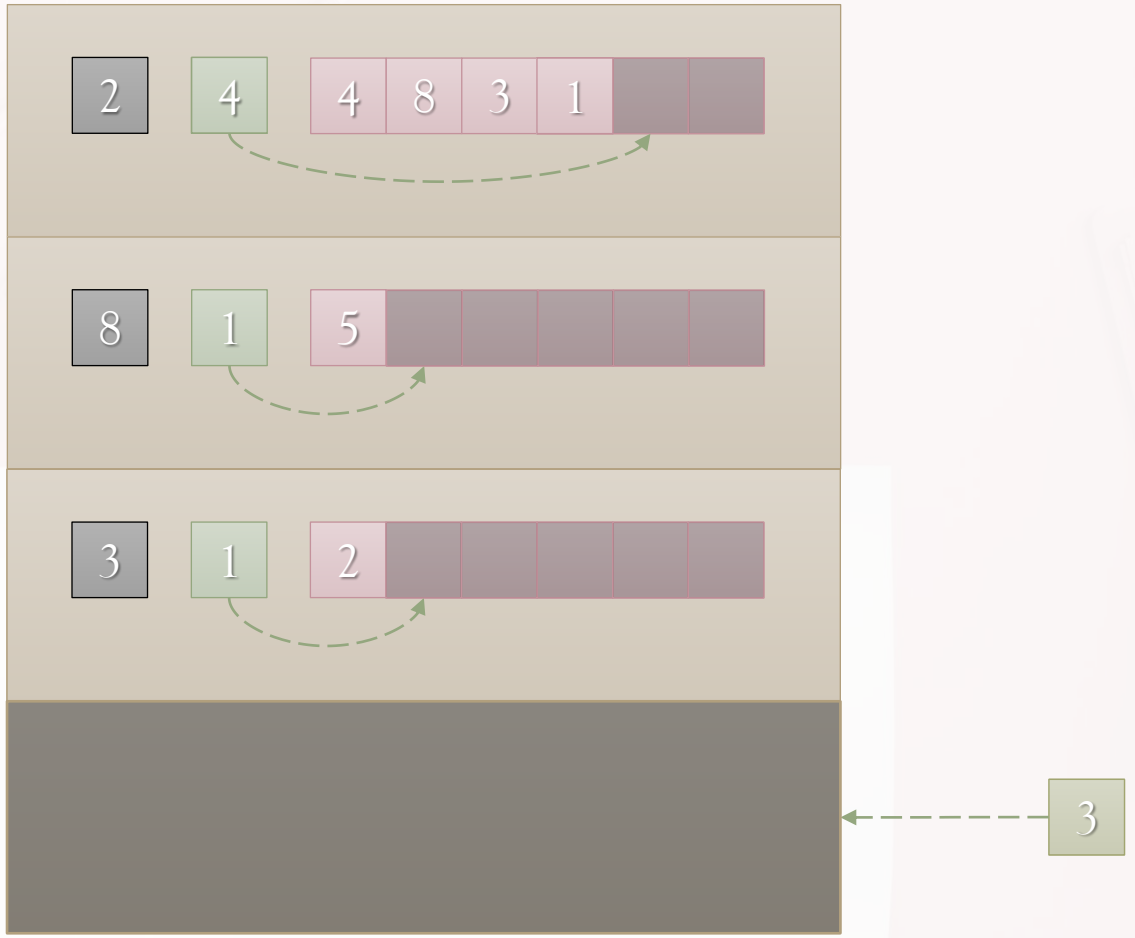
Diccionarios Múltiples - Implementación estática

agregar (3, 2)



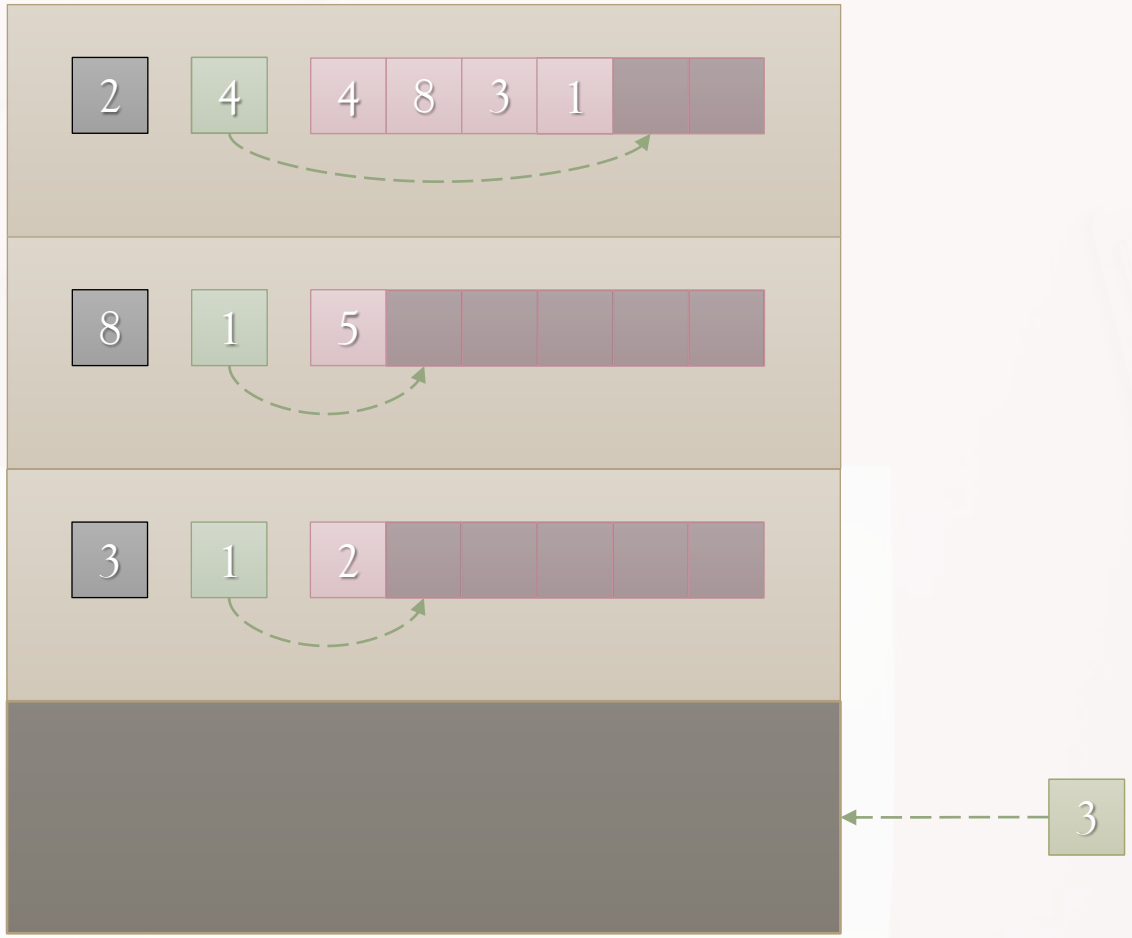
Diccionarios Múltiples - Implementación estática

agregar (3, 2)



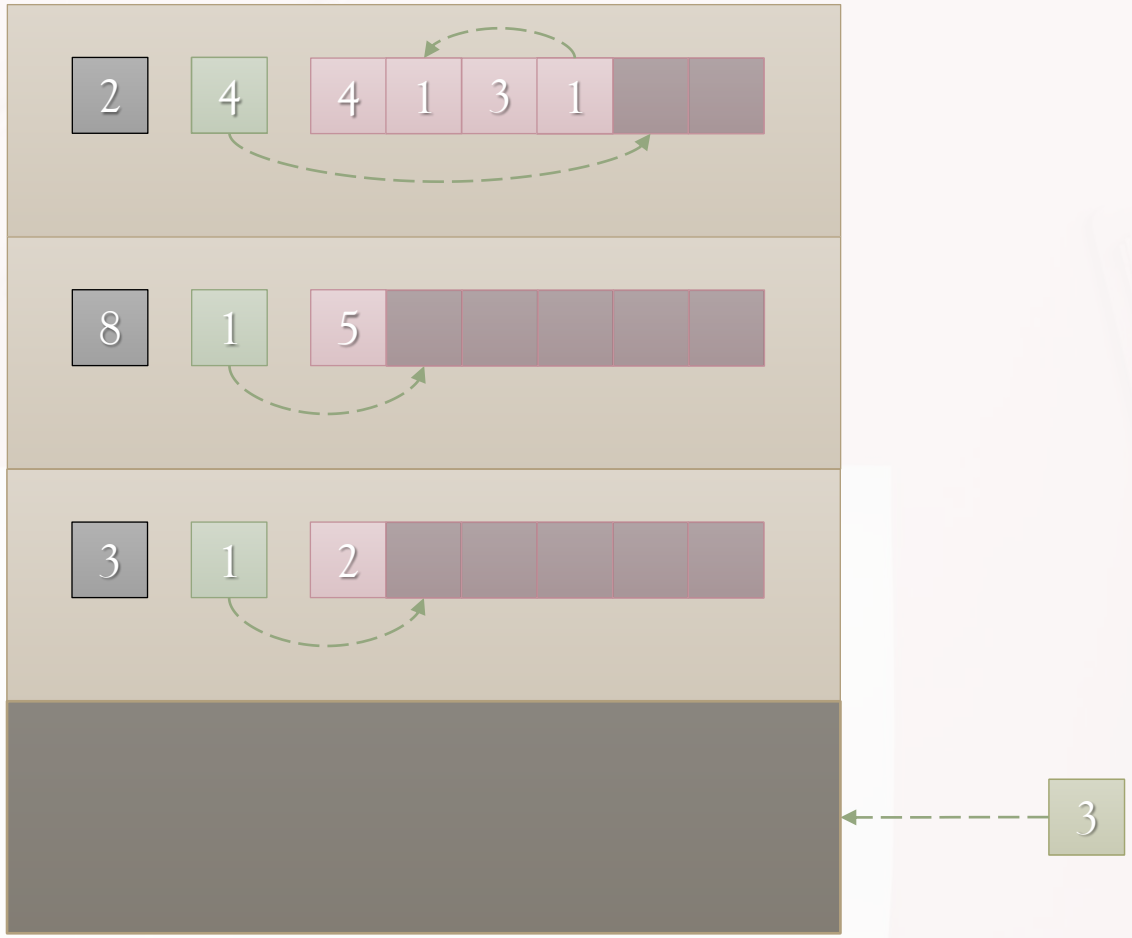
Diccionarios Múltiples - Implementación estática

eliminarValor (2, 8)



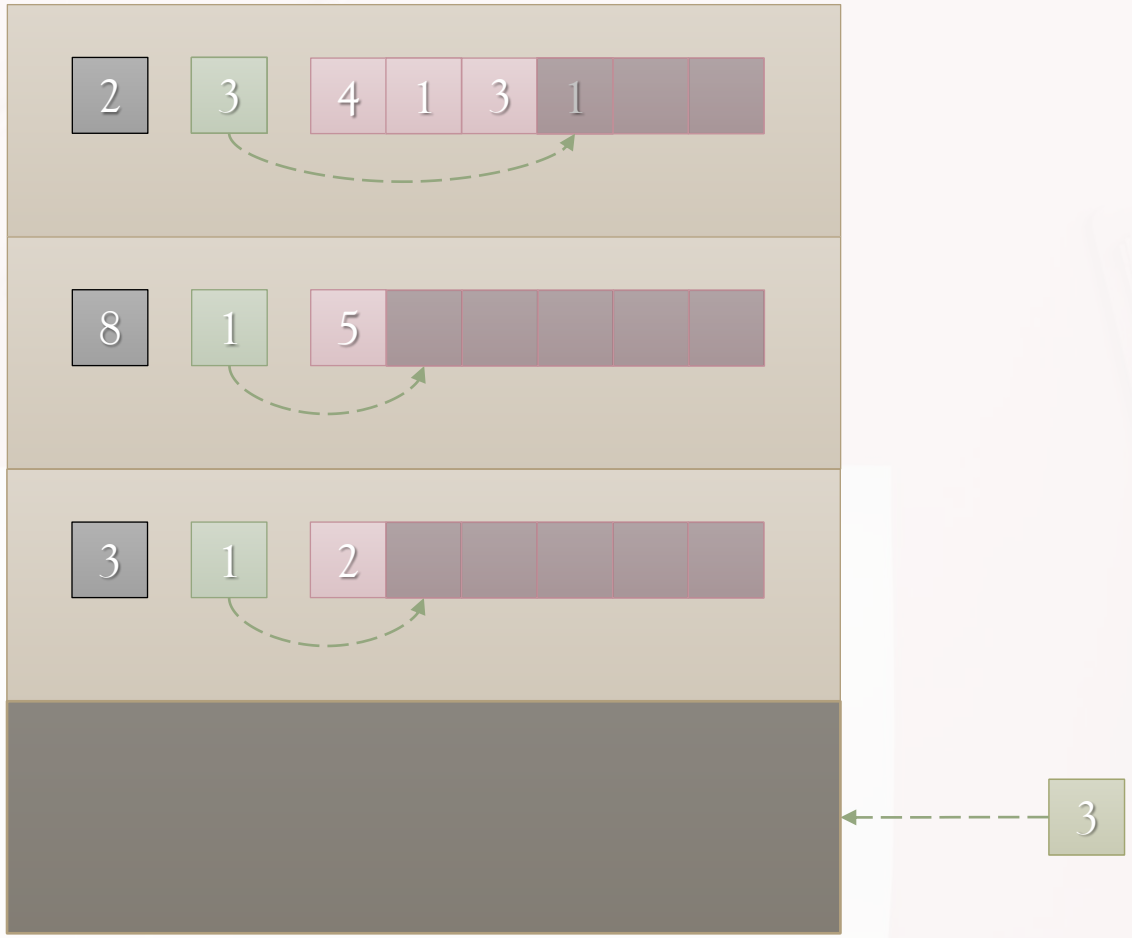
Diccionarios Múltiples - Implementación estática

eliminarValor (2, 8)



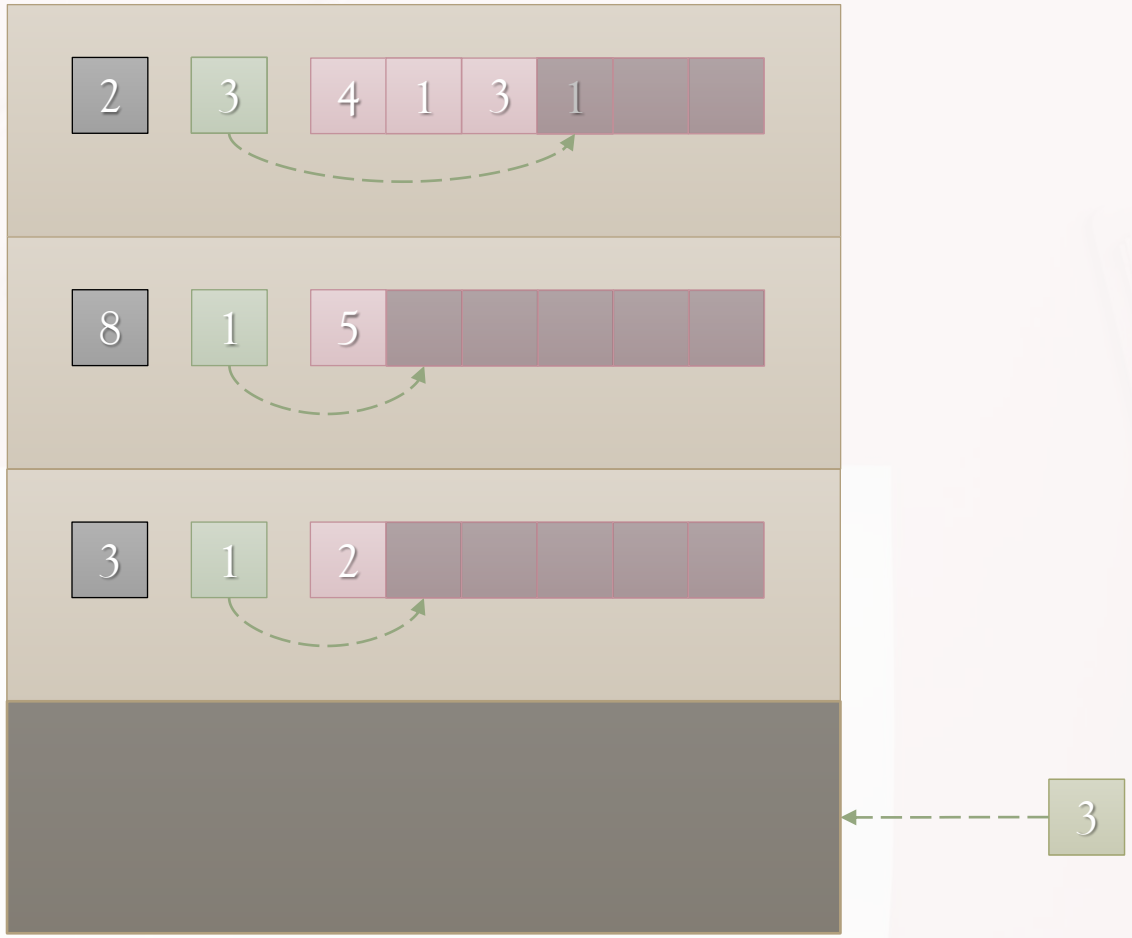
Diccionarios Múltiples - Implementación estática

eliminarValor (2, 8)



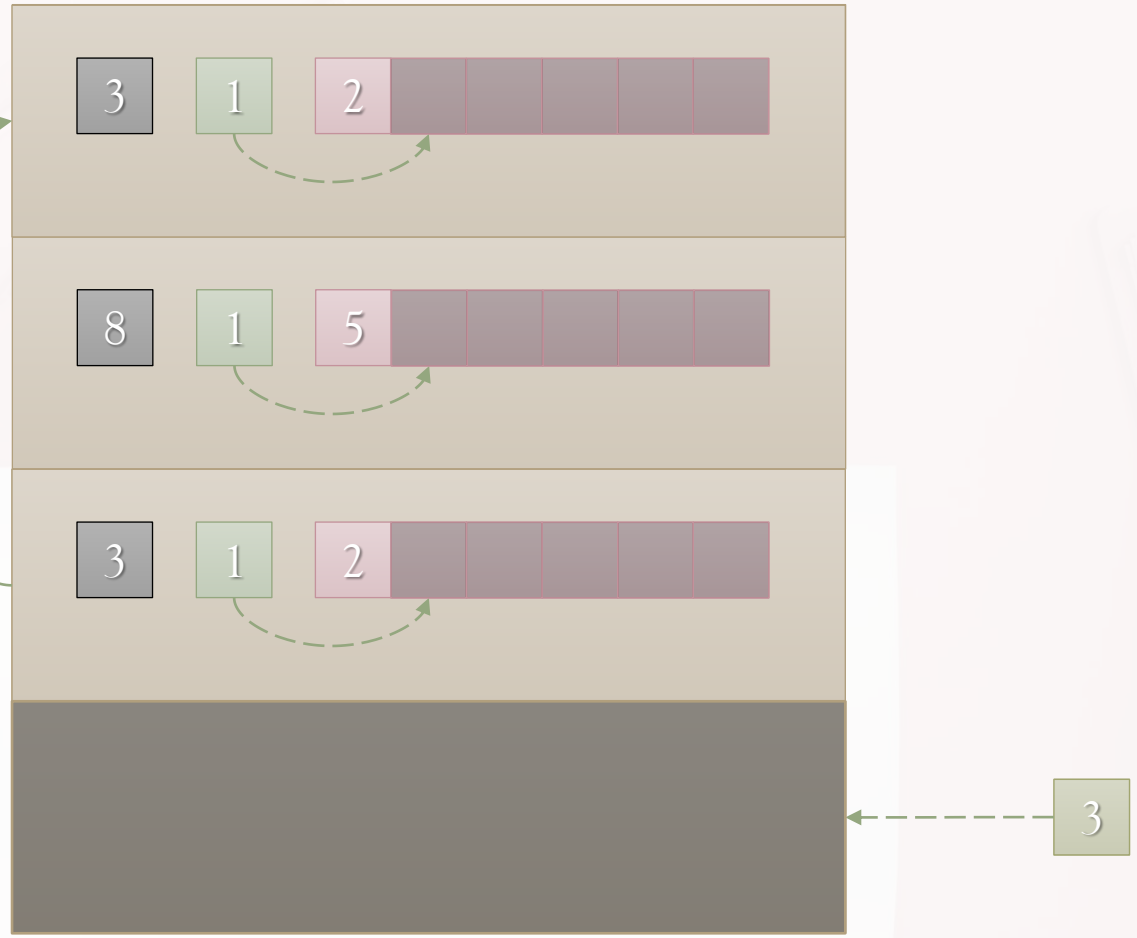
Diccionarios Múltiples - Implementación estática

eliminar (2)



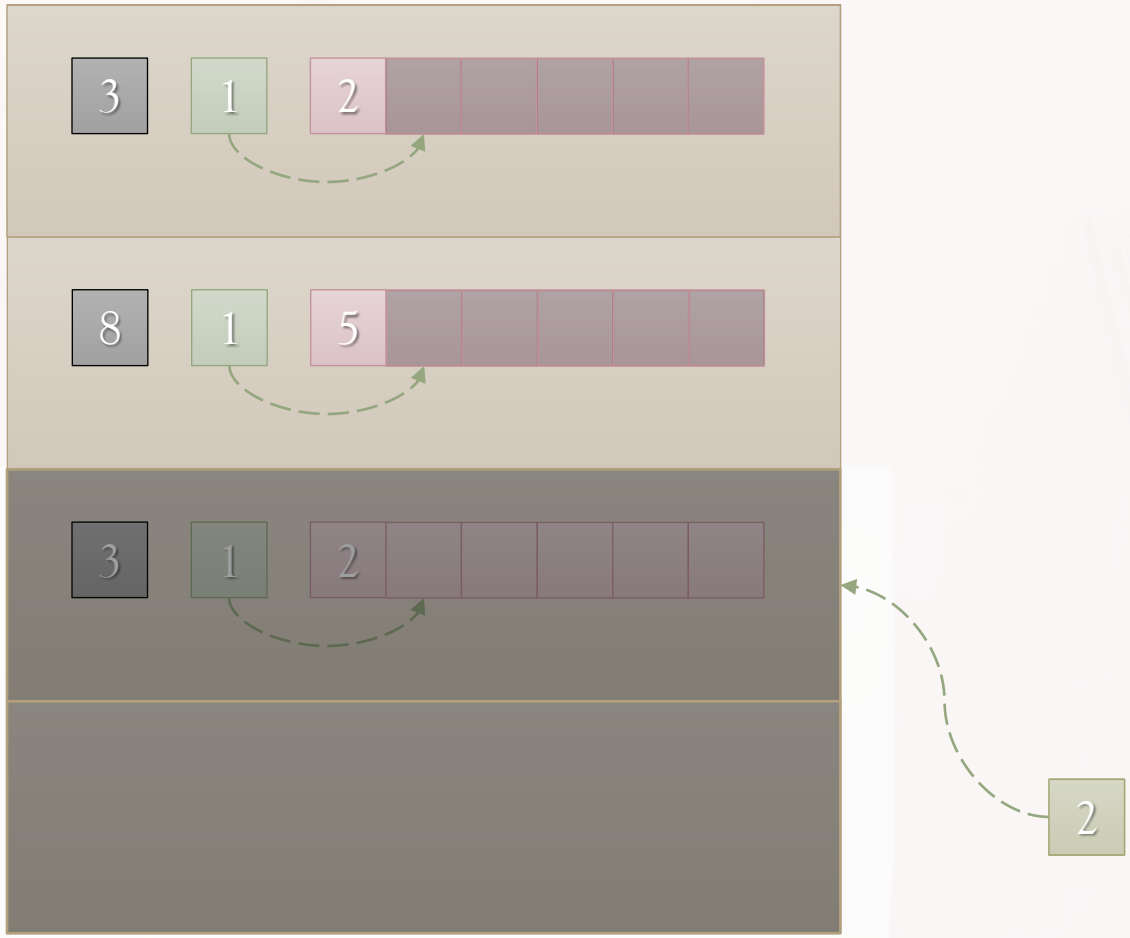
Diccionarios Múltiples - Implementación estática

eliminar (2)



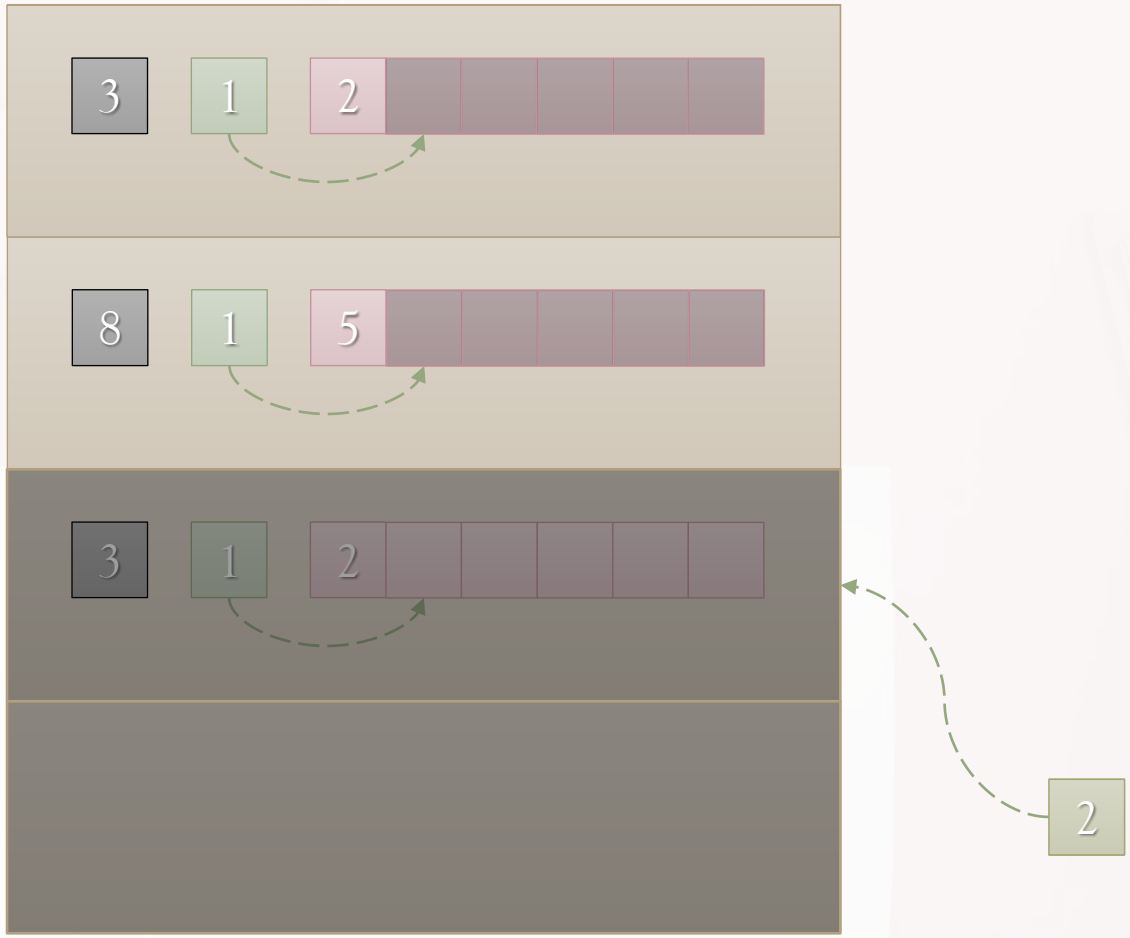
Diccionarios Múltiples - Implementación estática

eliminar (2)



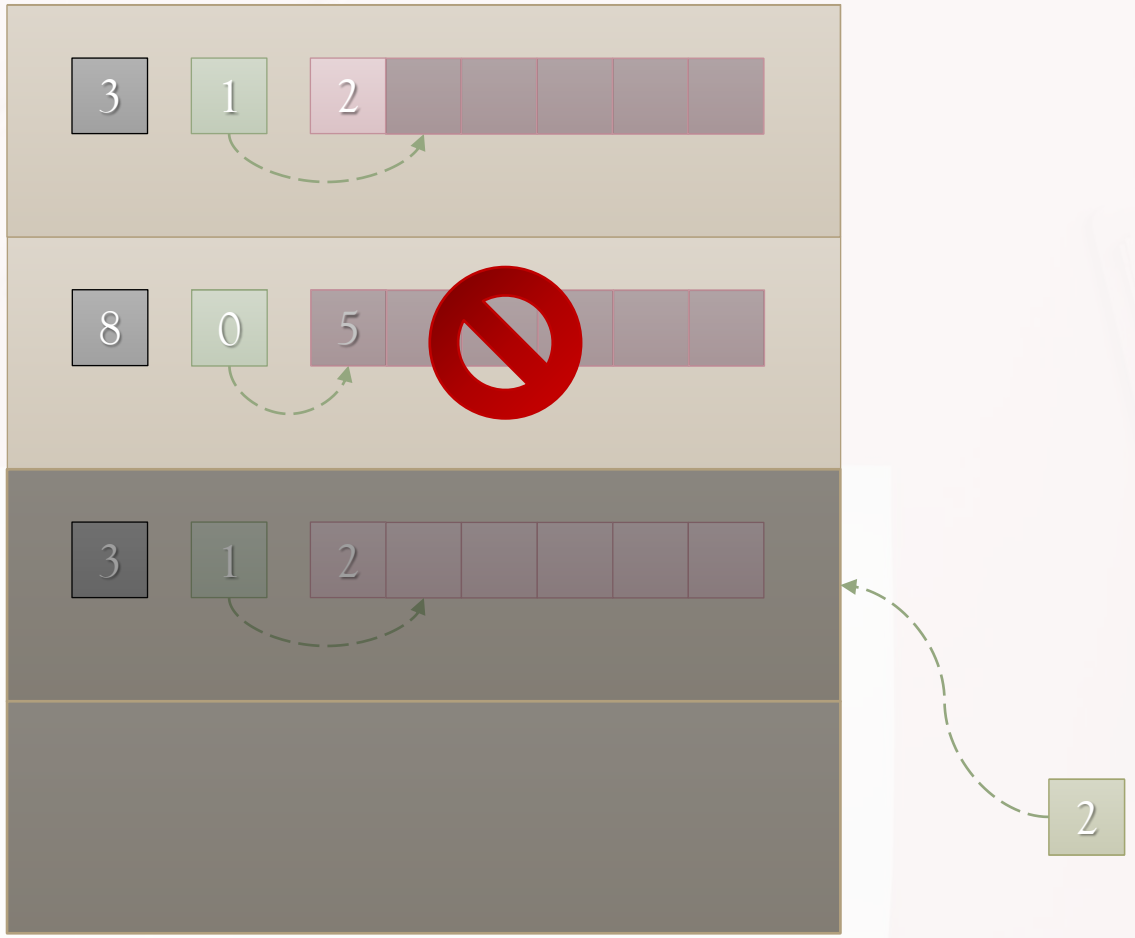
Diccionarios Múltiples - Implementación estática

eliminarValor (8, 5)



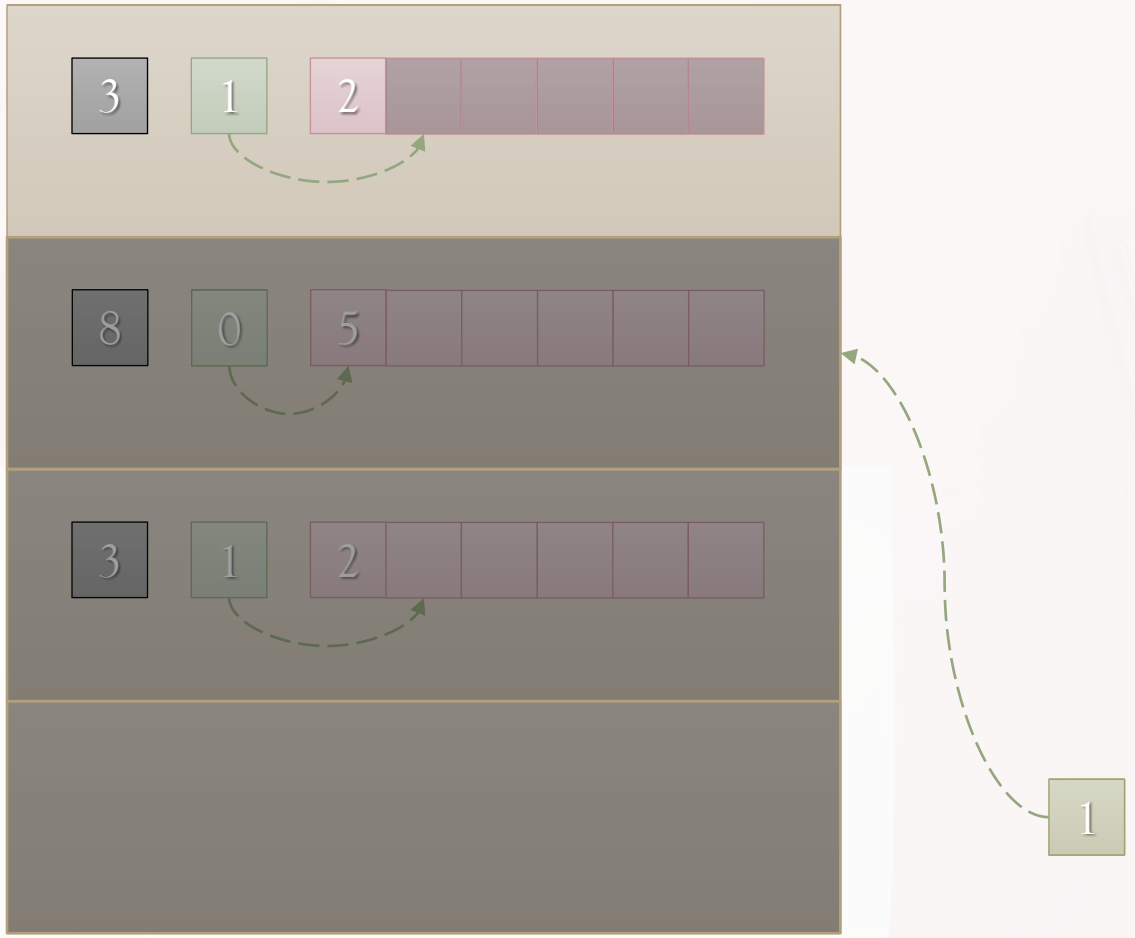
Diccionarios Múltiples - Implementación estática

eliminarValor (8, 5)



Diccionarios Múltiples - Implementación estática

eliminarValor (8, 5)



Implementación estática Aclaraciones

- Los métodos privados ***clave2Indice*** y ***valor2Indice*** devuelven respectivamente la posición de una clave en el arreglo de elementos y la posición de un valor para un valor e dado en el arreglo e.valores. Si la clave o el valor no existen, se devuelve -1.
- Tanto el vector elementos, como el entero cantClaves y los métodos clave2Indice y valor2Indice ***no son accesibles desde afuera de la implementación*** (son privados).

Implementación estática
Aclaraciones

- La ***eliminación*** de un elemento del vector elementos se representa dejándolo afuera de la parte del arreglo delimitada por la variable cantClaves; a los efectos prácticos, cualquier elemento elementos[i] situado en una posición $i \geq$ cantClaves ***no existe más en el diccionario***.


¡Muchas Gracias!

Bibliografía

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