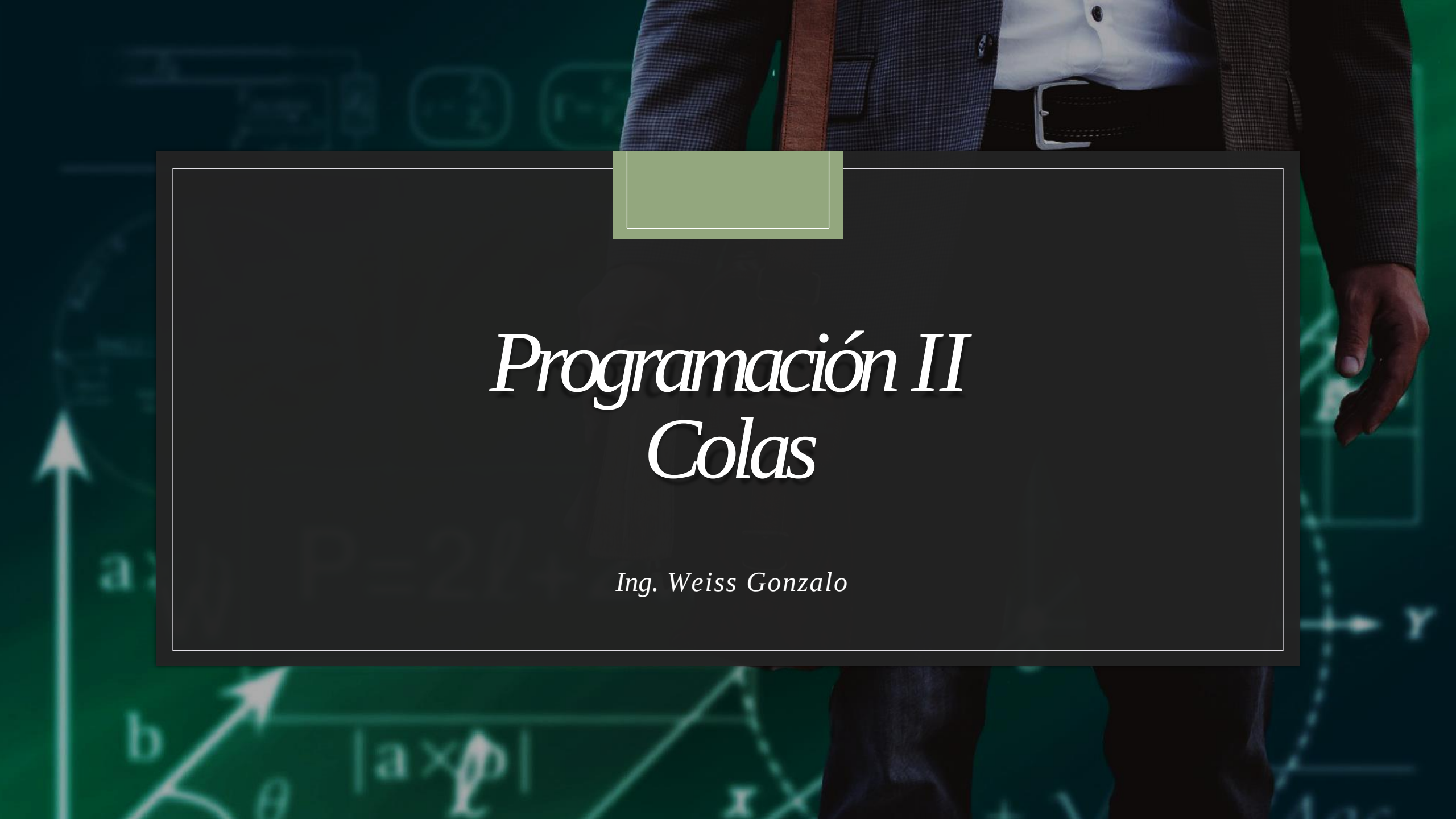


Programación II
Colas

Ing. Weiss Gonzalo

Temario

- *TDA*
- *Cola*
- *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.

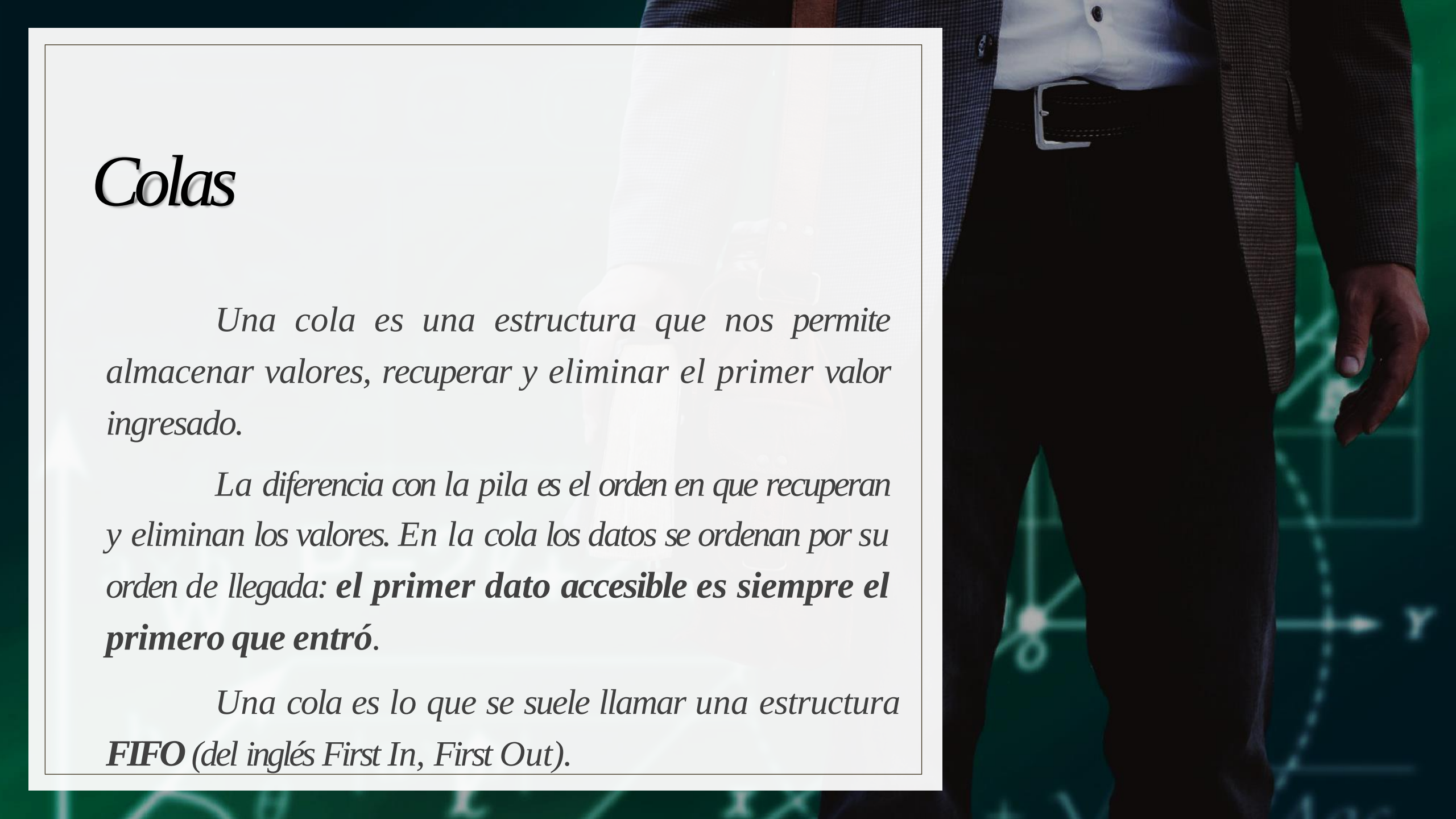
Colas

Colas

Una cola es una estructura que nos permite almacenar valores, recuperar y eliminar el primer valor ingresado.

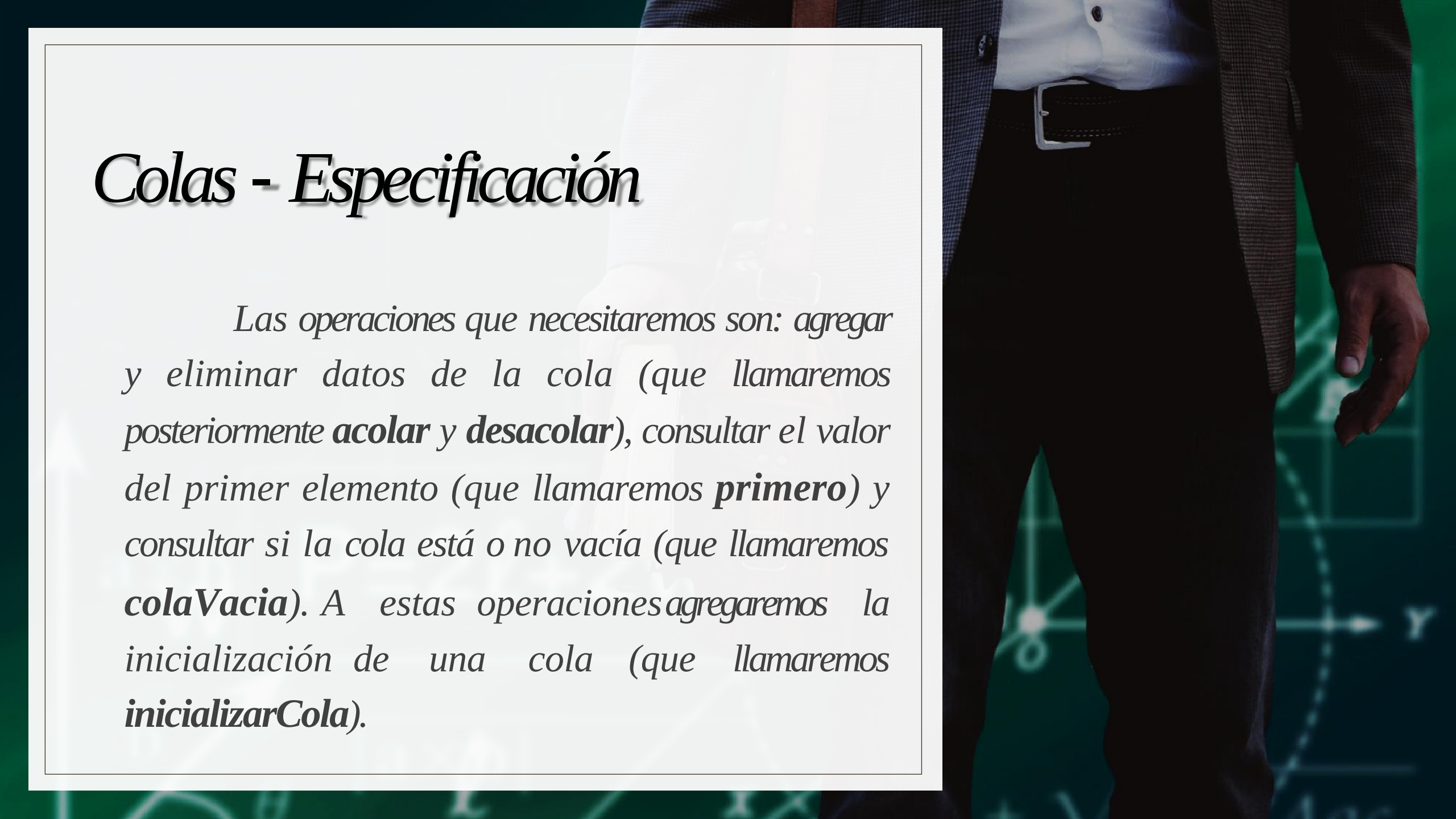
La diferencia con la pila es el orden en que recuperan y eliminan los valores. En la cola los datos se ordenan por su orden de llegada: ***el primer dato accesible es siempre el primero que entró****.*

Una cola es lo que se suele llamar una estructura ***FIFO*** *(del inglés First In, First Out).*



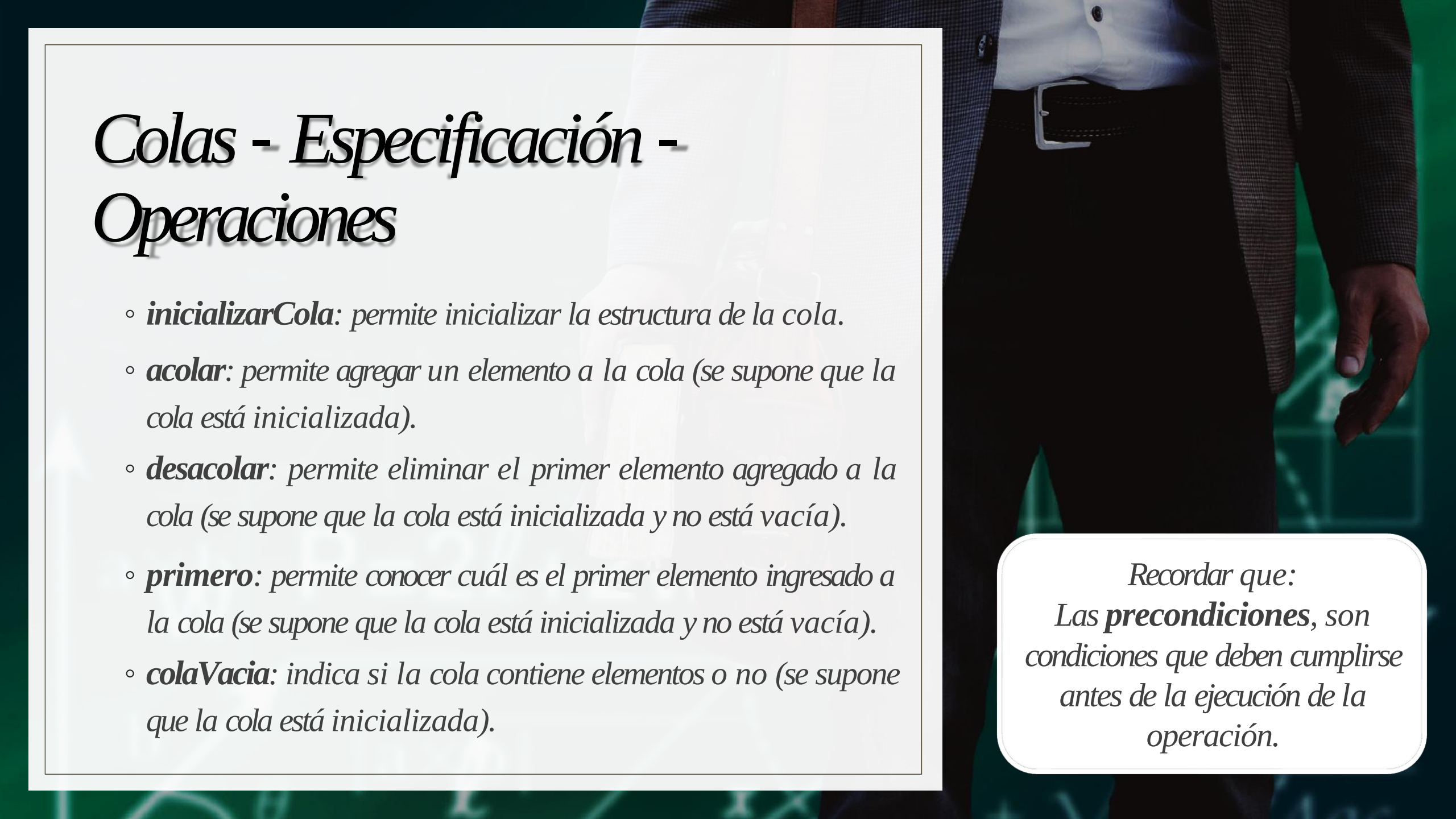
Colas - Especificación

Las operaciones que necesitaremos son: agregar y eliminar datos de la cola (que llamaremos posteriormente ***acolar*** *y* ***desacolar****), consultar el valor del primer elemento (que llamaremos* ***primero****) y consultar si la cola está o no vacía (que llamaremos* ***colaVacia****). A estas operaciones agregaremos la inicialización de una cola (que llamaremos* ***inicializarCola****).*



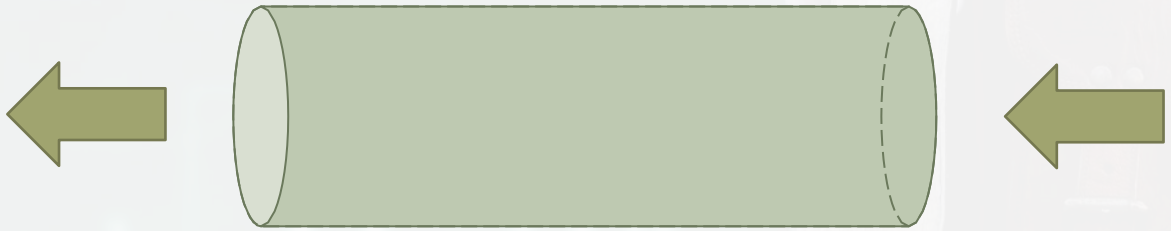
Colas - Especificación - Operaciones

- ***inicializarCola***: *permite inicializar la estructura de la cola.*
- ***acolar***: *permite agregar un elemento a la cola (se supone que la cola está inicializada).*
- ***desacolar***: *permite eliminar el primer elemento agregado a la cola (se supone que la cola está inicializada y no está vacía).*
- ***primero***: *permite conocer cuál es el primer elemento ingresado a la cola (se supone que la cola está inicializada y no está vacía).*
- ***colaVacia***: *indica si la cola contiene elementos o no (se supone que la cola está inicializada).*



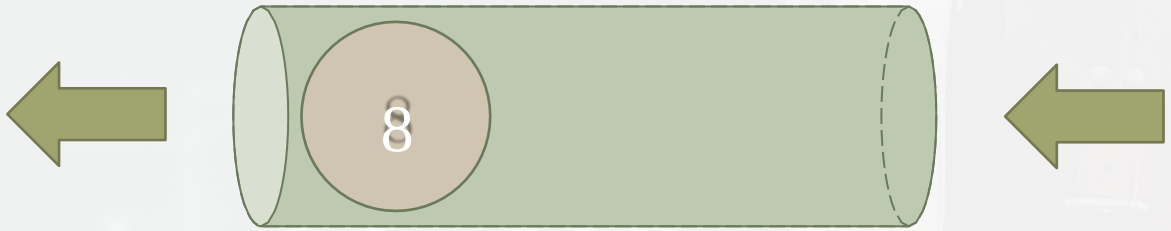
Recordar que:
Las ***precondiciones****, son condiciones que deben cumplirse antes de la ejecución de la operación.*

Colas - Especificación



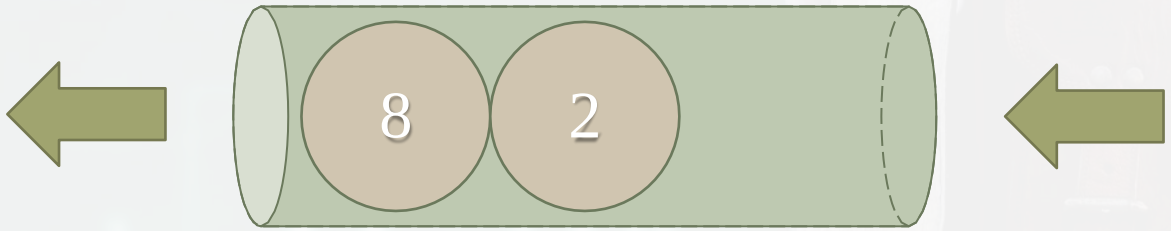
colaVacia () = true

Colas - Especificación



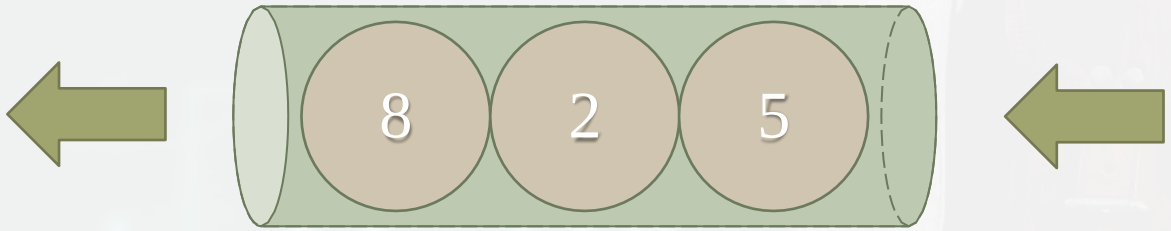
acolar (8)
colaVacia () = false
primero () = 8

Colas - Especificación



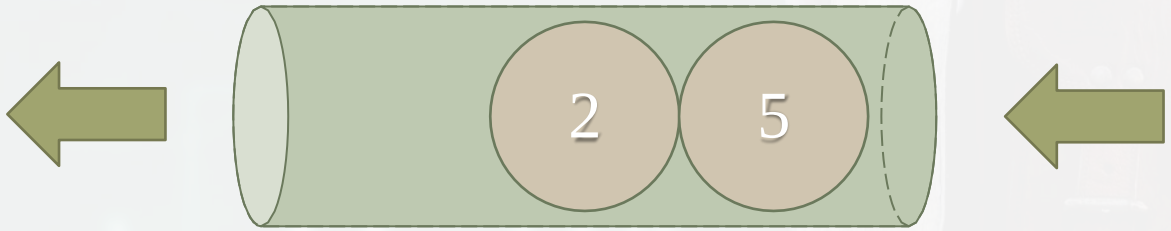
acolar (2)
colaVacia () = false
primero () = 8

Colas - Especificación



acolar (3)
colaVacia () = false
primero () = 8

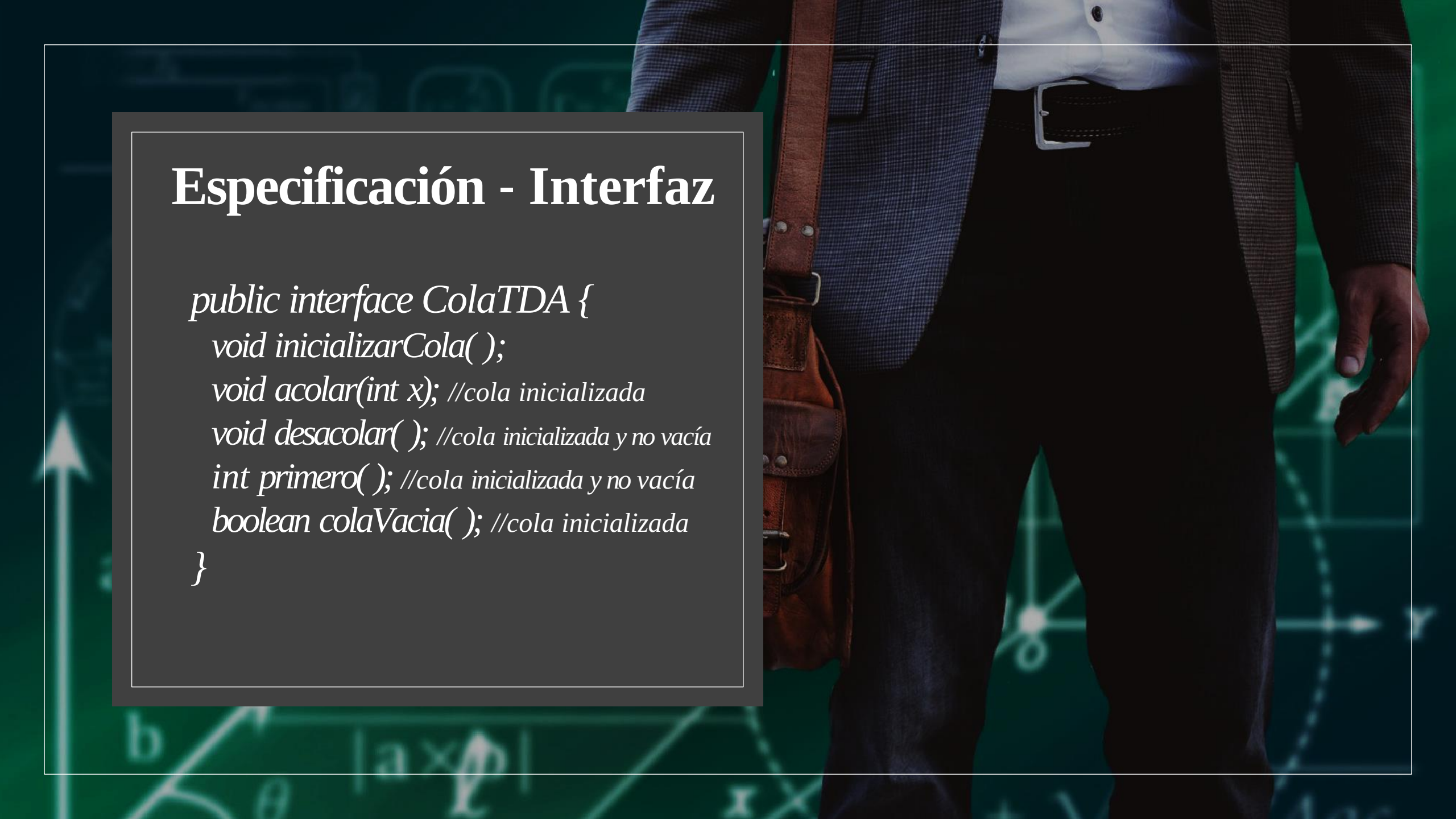
Colas - Especificación

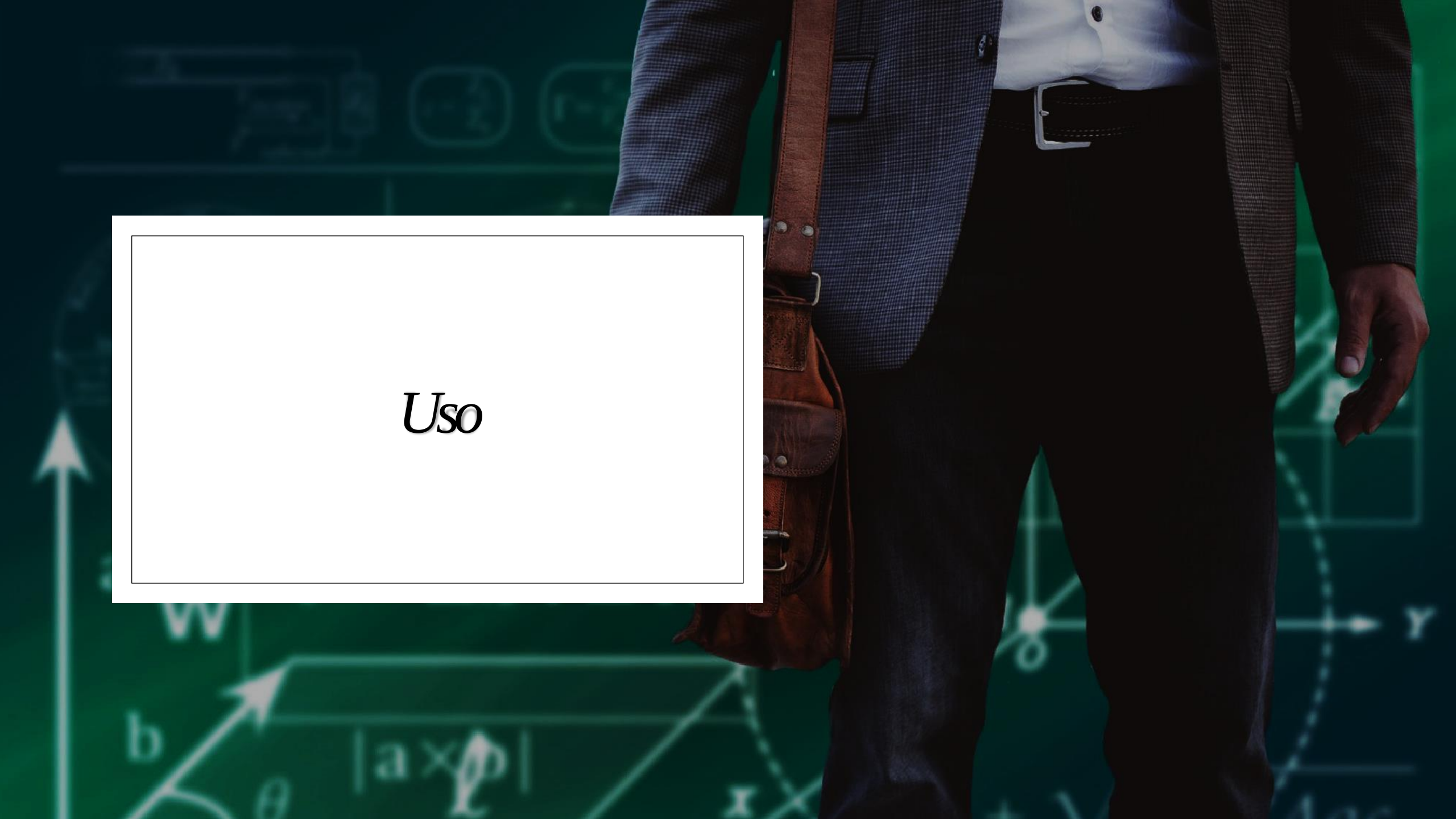


encolar (3)
encolar (2)
encolar (5)
colaVacia () = false
primero () = 2

Especificación - Interfaz

```
public interface ColaTDA {
   void inicializarCola( );
   void acolar(int x); //cola inicializada
   void desacolar( ); //cola inicializada y no vacía
   int primero( ); //cola inicializada y no vacía
   boolean colaVacia( ); //cola inicializada
}
```




Uso

Cola - Uso - Ejemplos

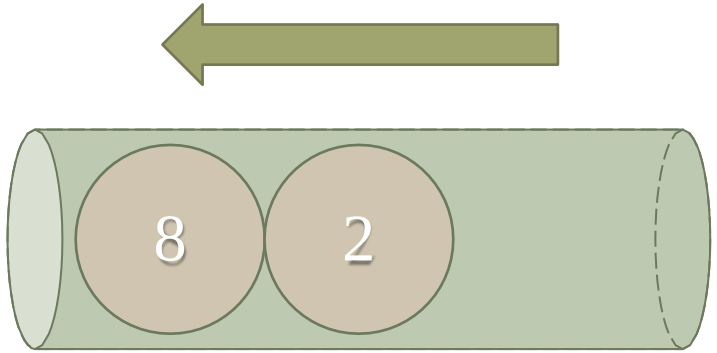
- *Vamos a escribir un método que nos permita pasar los elementos de una cola a otra. Los elementos en la cola destino quedarán en el mismo orden que en la cola origen.*

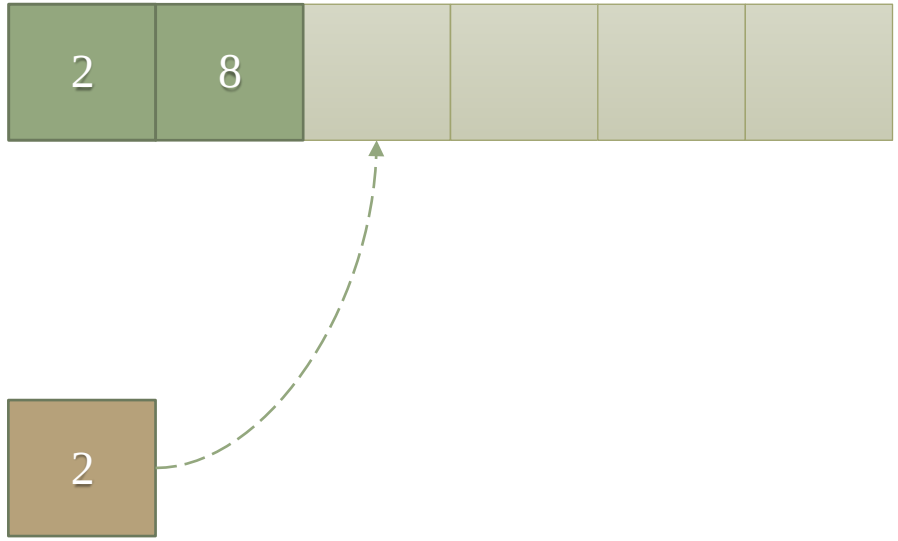
Implementación

Cola - Implementación estática Estrategia 1

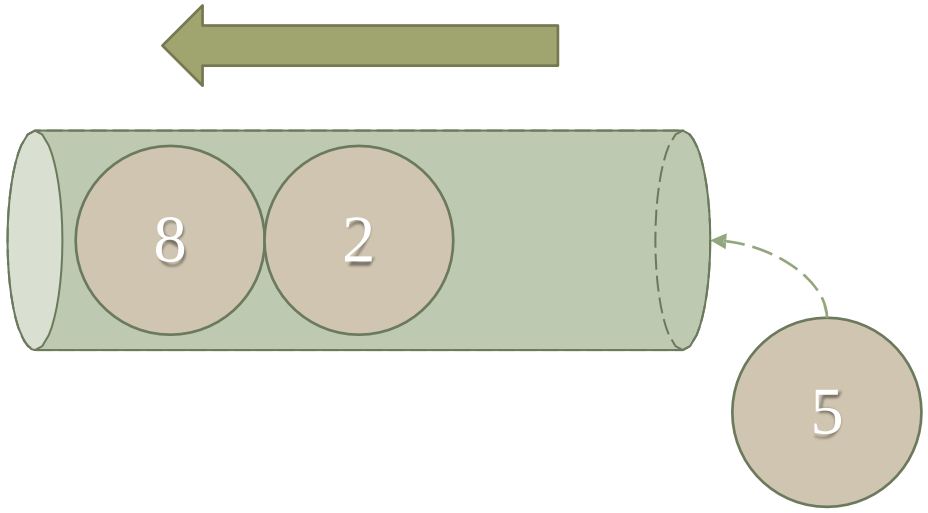
- *Se guardan los datos en un **arreglo** y además se tiene una **variable** que indica la cantidad de elementos que se tienen guardados en la cola.*
- *Cuando agregamos un nuevo elemento a la cola, el mismo se **guarda** en la posición cero del arreglo, por lo cual se requiere previo a la inserción un corrimiento a derecha de los elementos que se encuentran en la cola. Luego, la variable que me indica la cantidad de elementos se incrementa en uno.*
- *Cuando se tiene que **desacolar** un elemento de la cola, solo es necesario decrementar en una unidad esta variable (borrado lógico).*

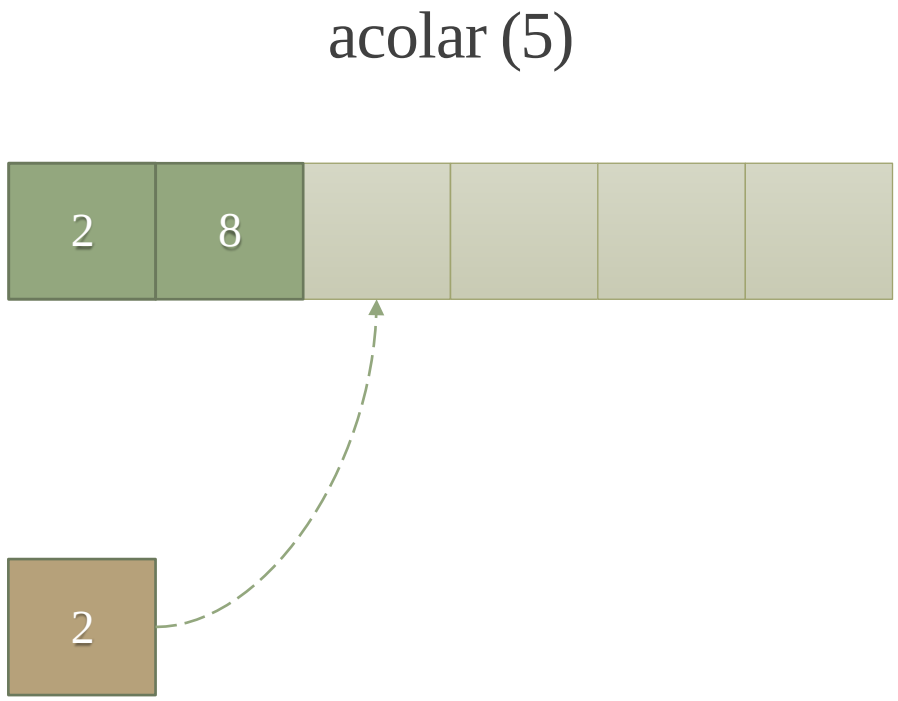
Cola - Implementación estática - Estrategia 1



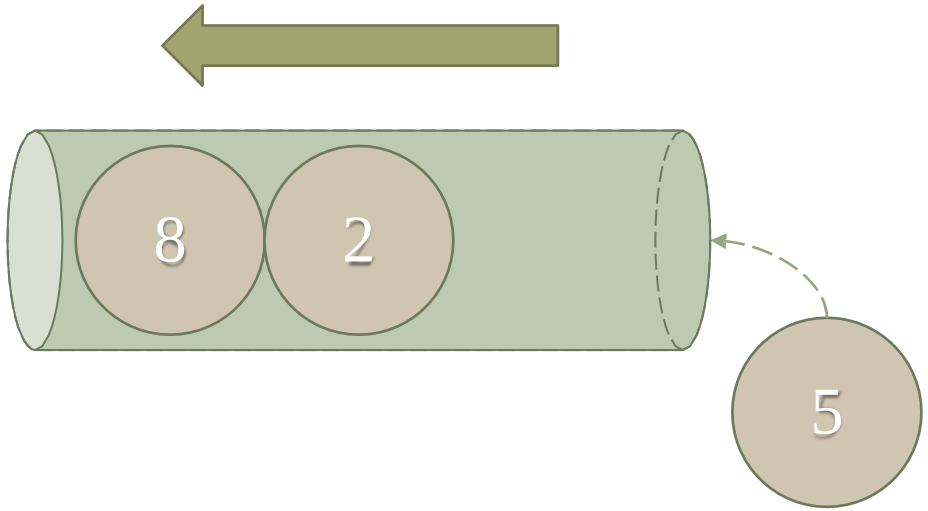


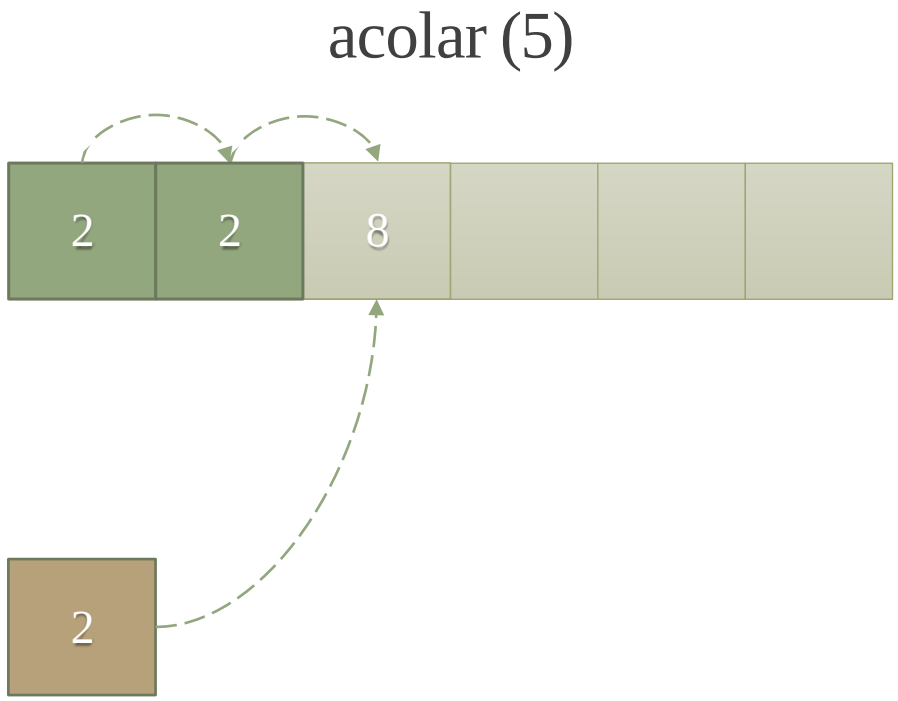
Cola - Implementación estática - Estrategia 1



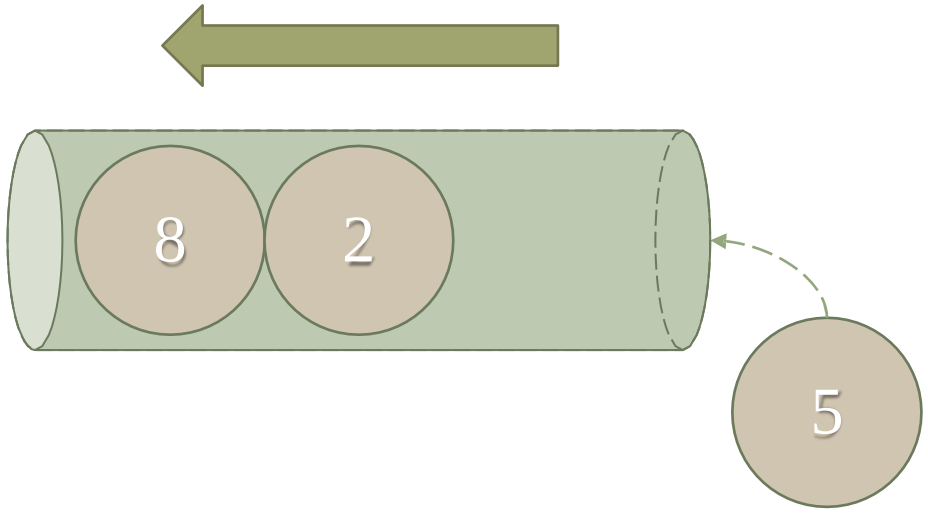


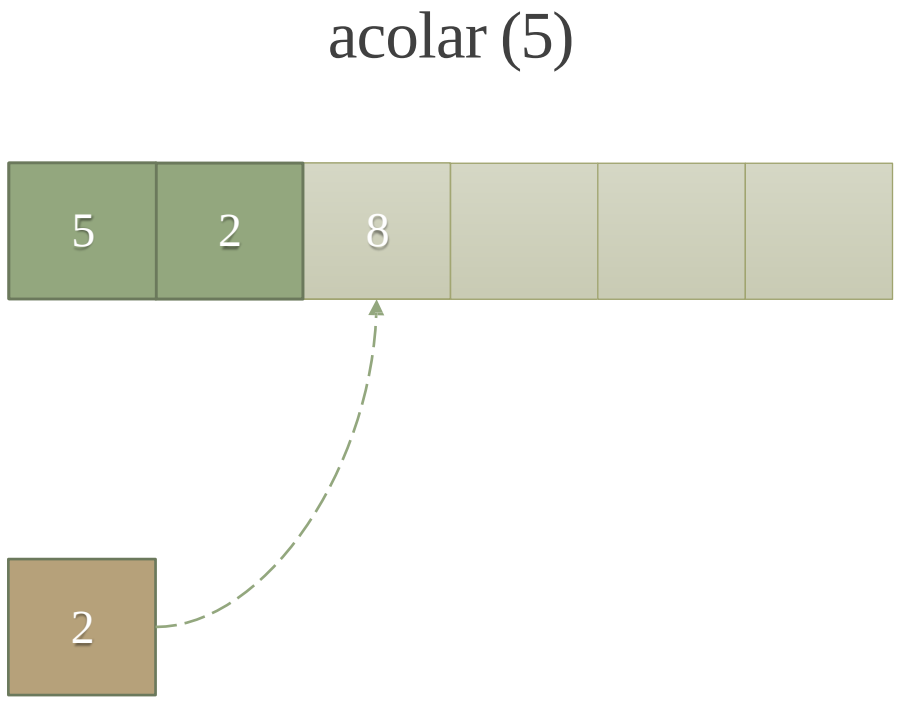
Cola - Implementación estática - Estrategia 1



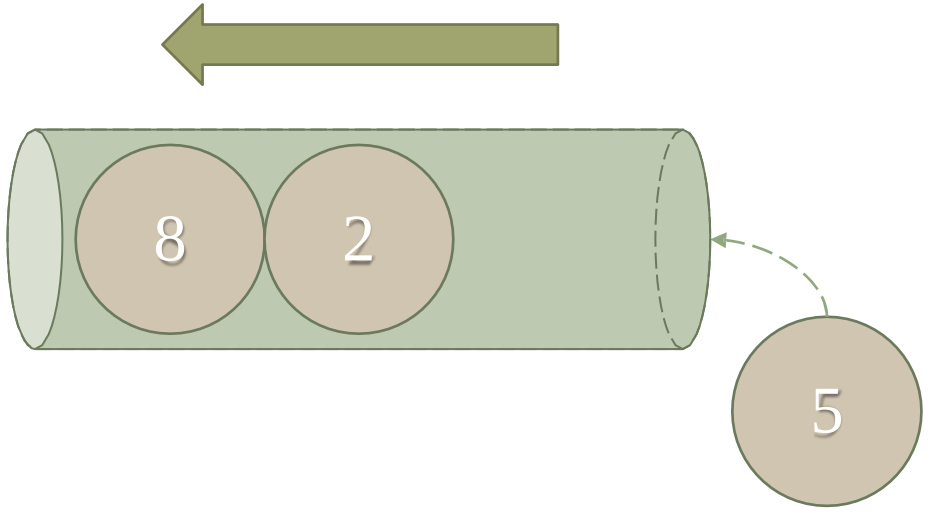


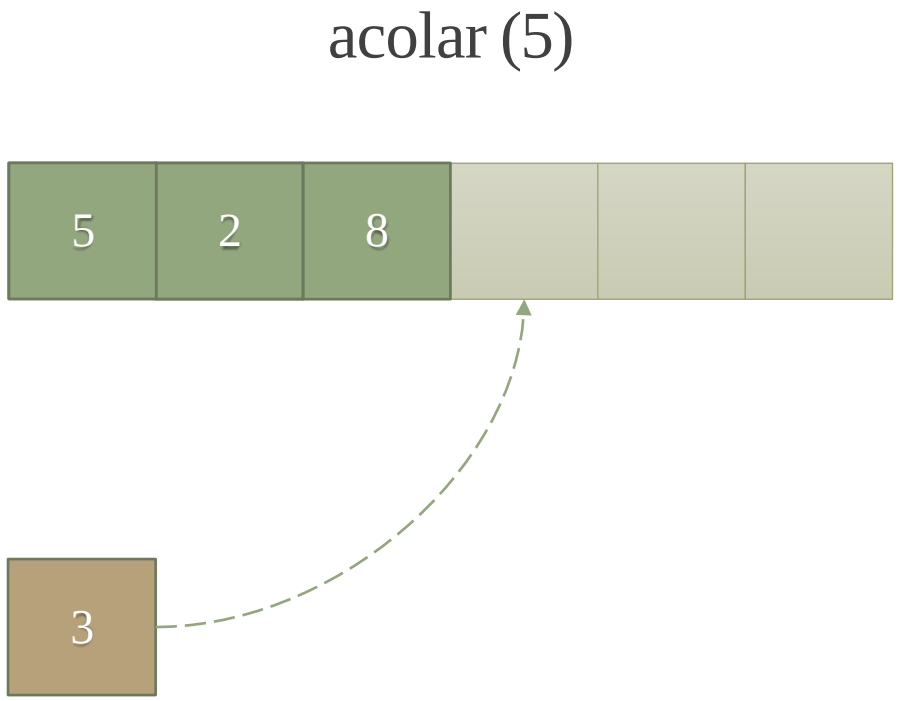
Cola - Implementación estática - Estrategia 1



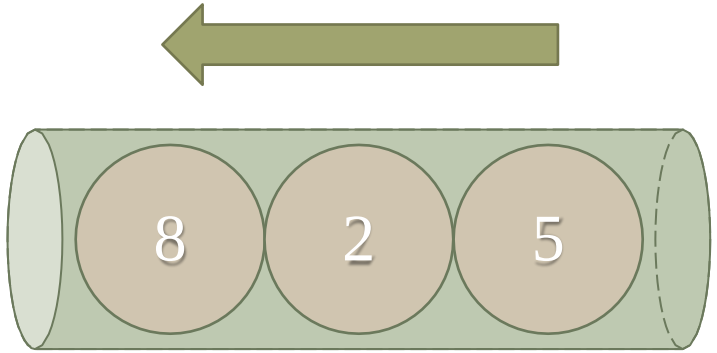


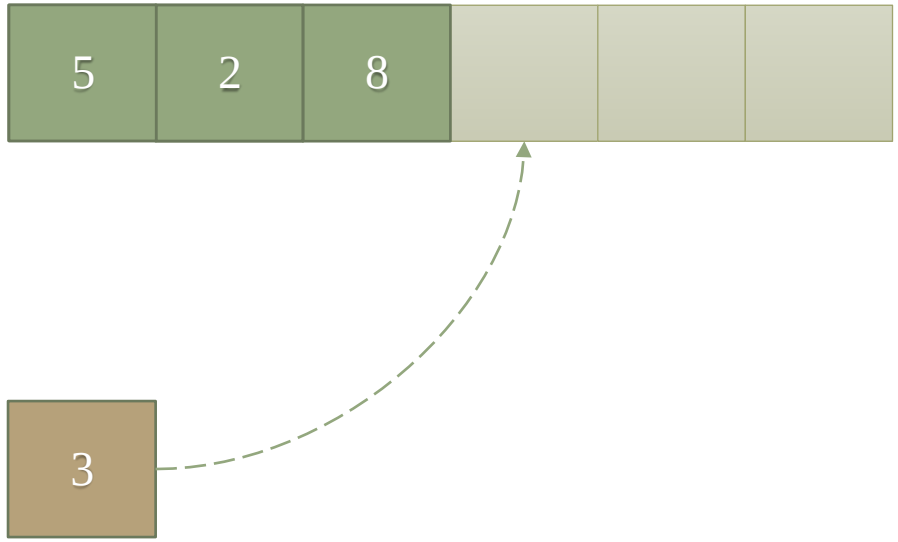
Cola - Implementación estática - Estrategia 1



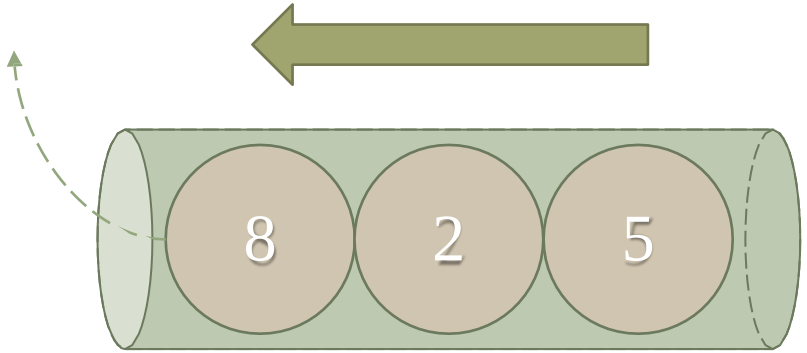


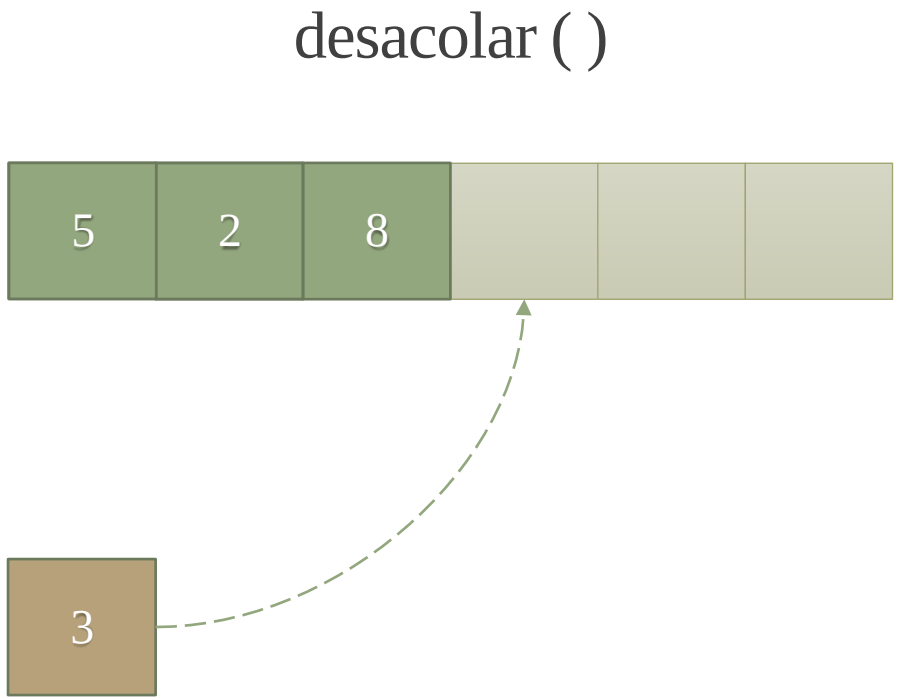
Cola - Implementación estática - Estrategia 1



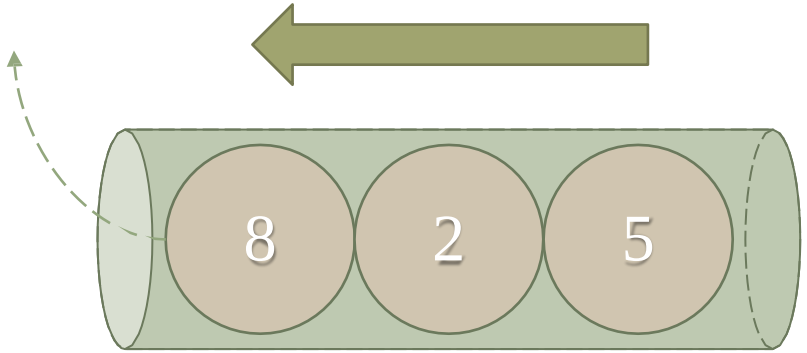


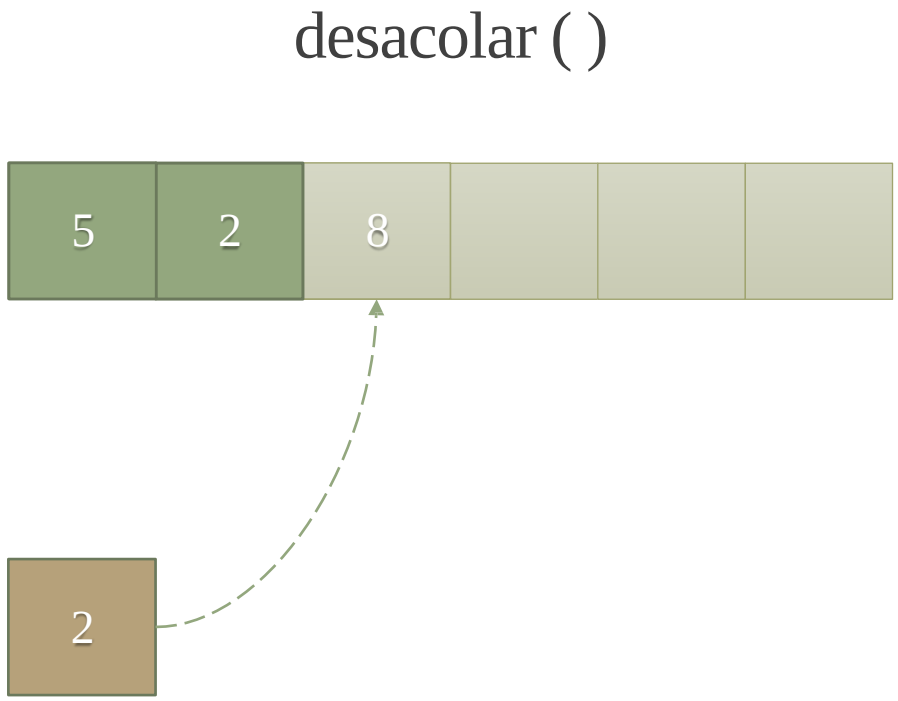
Cola - Implementación estática - Estrategia 1



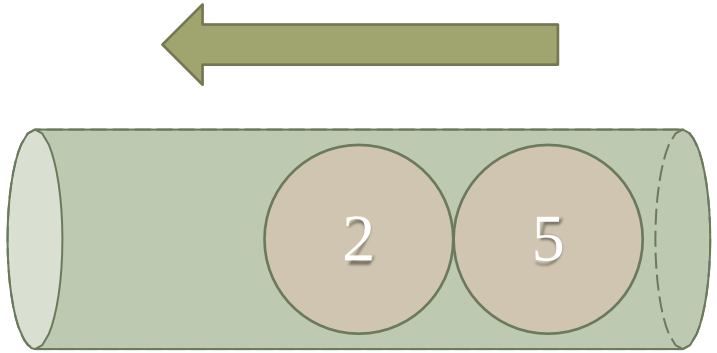


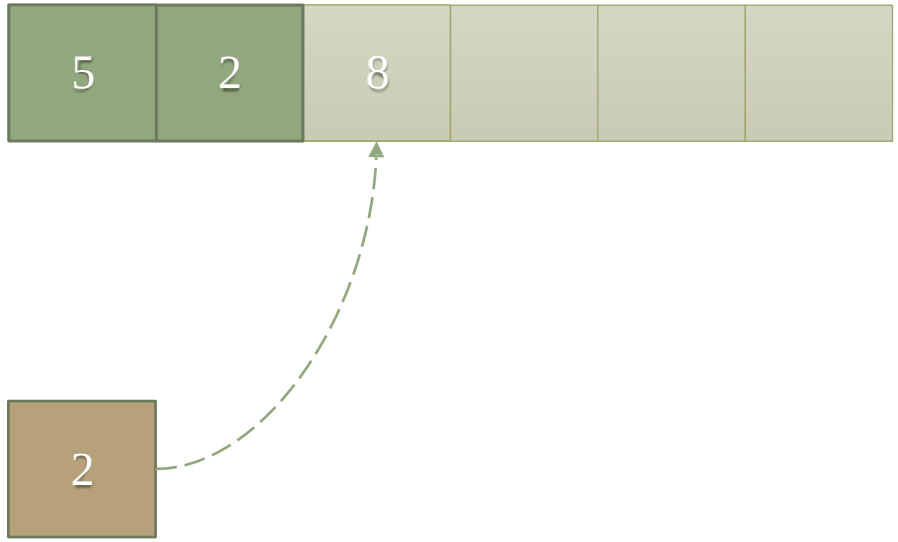
Cola - Implementación estática - Estrategia 1





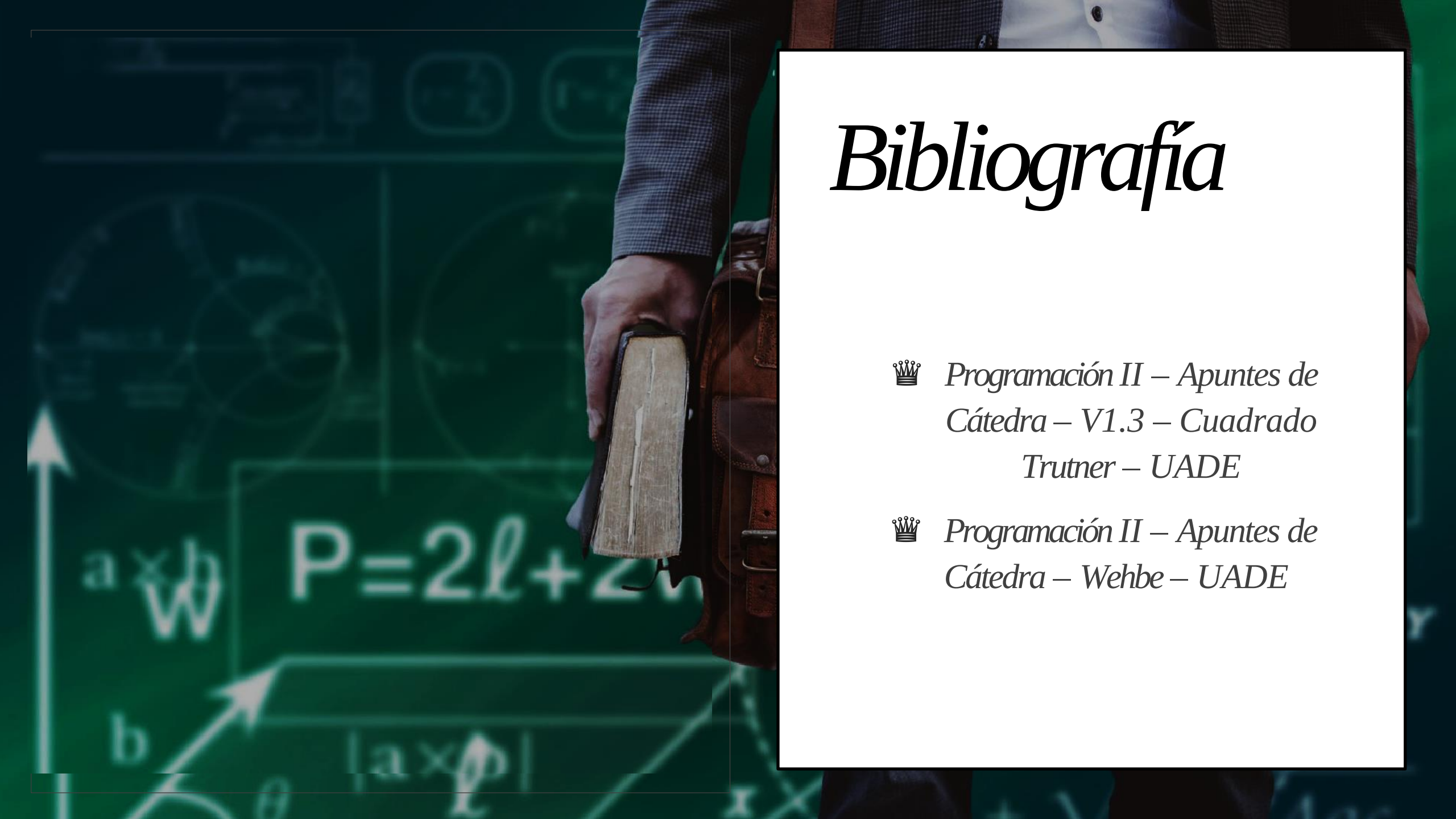
Cola - Implementación estática - Estrategia 1





Cola - 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 la cola***.
- *Tanto el vector arr, como el entero índice* ***no son accesibles desde afuera de la implementación*** *(son privados).*



Bibliografía

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