Estructura de Datos, Unahur

**Estructuras Pilas y Colas**

Prof. Sergio Gonzalez

Prof. Ariel Clocchiatti

**Estructura Pila:**

Una **pila** (***stack*** en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s)) es una [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos) que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo [LIFO](https://es.wikipedia.org/wiki/LIFO) (del inglés *Last In, First Out*, «último en entrar, primero en salir») . Esta estructura se aplica en multitud de supuestos en el área de [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica) debido a su simplicidad y capacidad de dar respuesta a numerosos procesos.

Para el manejo de los datos cuenta con dos operaciones básicas: **apilar (*push)***, que coloca un objeto en la pila, y su operación inversa, **desapilar** (***pop***), que retira el último elemento apilado.

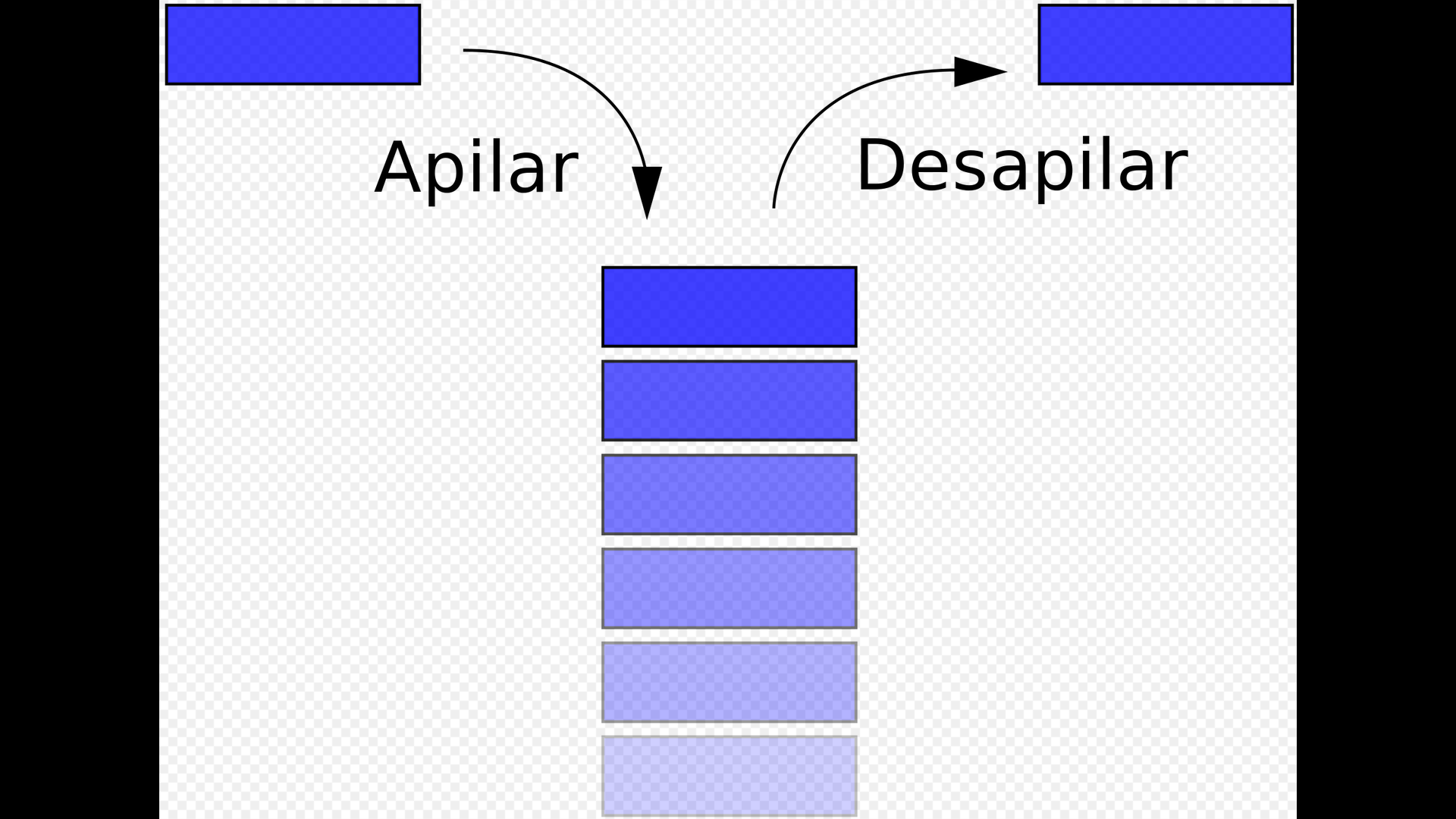
En cada momento sólo se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado. La operación **desapilar** (***pop***) permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al anterior (apilado con anterioridad), que pasa a ser el nuevo último.

Las **pilas** suelen emplearse en los siguientes contextos:

* Evaluación de expresiones en [notación postfija](https://es.wikipedia.org/wiki/Notaci%C3%B3n_postfija) ([notación polaca inversa](https://es.wikipedia.org/wiki/Notaci%C3%B3n_polaca_inversa)).
* Reconocedores sintácticos de [lenguajes independientes del contexto](https://es.wikipedia.org/w/index.php?title=Lenguaje_libre_de_contexto&action=edit&redlink=1).
* Implementación de [recursividad](https://es.wikipedia.org/wiki/Recursividad) (llamadas a funciones recursivas).
* En un [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) cada proceso tiene un espacio de memoria (pila) para almacenar valores y llamadas a funciones.

Existen Pilas como estructuras dinámicas y también como estructuras estáticas (de tamaño fijo).

El siguiente grafico modeliza una Pila:



**Pila como tipo abstracto de datos**

A modo de resumen, la pila es un contenedor de nodos y tiene dos operaciones básicas: **push** (apilar) y **pop** (desapilar). «Push» añade un nodo a la parte superior de la pila, dejando por debajo el resto de los nodos ya presentes en la pila. «Pop**»** devuelve y elimina el actual nodo superior de la pila.

El concepto de Pila es el mismo que ya conocemos cuando tenemos una Pila de ropa almacenada en una valija o una pila de platos en la cocina. Cuando lavamos los platos, tomamos el plato del tope de la pila y a medida que los vamos lavando, los estamos desapilando de la pila. Para llegar a la base de la pila, debemos ir lavando uno a uno los platos (desapilando), hasta llegar al primer plato que pusimos en la pila. (El último plato puesto, es el primero en ser lavado LastInFirstOut).

### Operaciones

Habitualmente, junto a las dos operaciones básicas de apilar y desapilar (push, pop), las pilas puede implementar otra serie de funciones:

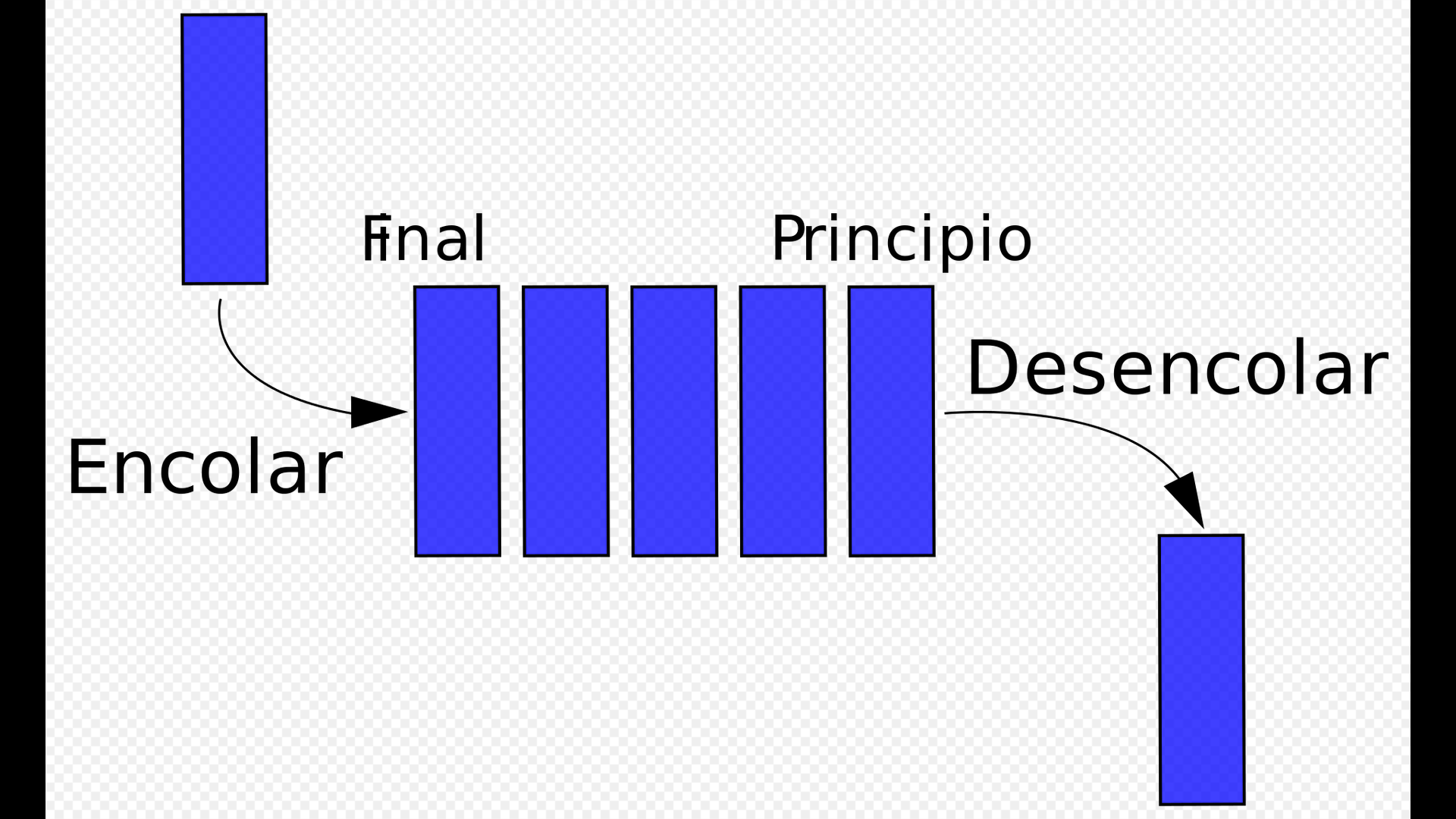
* **Crear (constructor):** crea la pila vacía.
* **Tamaño (*size*):** regresa el número de elementos de la pila.
* **Apilar (*push*):** añade un elemento a la pila.
* **Desapilar *(pop*):** lee y retira el elemento superior de la pila.
* **Leer** **último** (***top* o *peek***): lee el elemento superior de la pila sin retirarlo.
* **Vacía (*empty*):** devuelve ‘true’ si la pila está sin elementos o falso en caso de que contenga alguno.

Una Pila puede implementarse fácilmente, por ejemplo, con una lista dinámica o un array (estructura estática). Lo que identifica a una estructura de datos como una Pila es su interfaz: al usuario sólo se le permite colocar y extraer datos en el modo que se espera de una pila y algunas otras operaciones auxiliares. De ésta forma garantizamos la estructura LIFO.

**Estructura Cola**

Una **cola** es una [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos), caracterizada por ser una secuencia de elementos en la que la operación de inserción *push* se realiza por un extremo y la operación de extracción *pull* por el otro. La estructura de Cola es [FIFO](https://es.wikipedia.org/wiki/FIFO) (del inglés *First In First Out*), debido a que el primer elemento en entrar será también el primero en salir.

Las colas se utilizan en sistemas [informáticos](https://es.wikipedia.org/wiki/Inform%C3%A1tica), [transportes](https://es.wikipedia.org/wiki/Transporte) e investigación operativa (entre otros), donde los objetos, personas o eventos son tomados como datos que se almacenan y se guardan mediante colas para su posterior procesamiento. Este tipo de estructura de datos abstracta comúnmente se implementa en los lenguajes de programación como listas enlazadas.



La particularidad de una estructura de datos de tipo Cola es el hecho de que sólo podemos acceder al primer y al último elemento de la estructura. Así mismo, los elementos sólo se pueden eliminar por el principio y sólo se pueden añadir por el final de la cola.

Ejemplo, las impresoras tienen implementada una Cola de impresión, donde el primer documento que llega, es el primero en ser impreso.

Y cuando todos respetamos la cola del supermercado, el primer changuito que llega, es el primero en ser despachado.

**Operaciones**:

* **Crear**: se crea la cola vacía.
* **Encolar**: se añade un elemento a la cola. Se añade al final de esta.
* **Desencolar**: se elimina el elemento frontal de la cola, es decir, el primer elemento que entró.
* **Frente**: (consultar, front): se devuelve el elemento frontal de la cola, es decir, el primer elemento que entró.
* **Tamaño (size)**: devuelve la cantidad de elementos en la cola.
* **Vacía**: True or False si tiene o no elementos.