

Ejercicio de Clases Java donde usaremos Un interface y trabajaremos con excepciones implementando un elemento Pila de enteros.

Definición de Pila: Una **pila** (*stack* en [inglés](#)) es una lista ordenada o [estructura de datos](#) que permite almacenar y recuperar datos, el modo de acceso a sus elementos es de tipo [LIFO](#) (del inglés *Last In, First Out*, «último en entrar, primero en salir»).

Crear un interface llamada InterfaceDePila que declaro los siguientes métodos:

1. **estaVacía()** : devuelve true si la colección está vacía y false en caso contrario.
2. **extraer()** : devuelve y elimina el primer elemento de la colección.
3. **primero()** : devuelve el primer elemento de la colección.
4. **añadir()** : añade un objeto por el extremo que corresponda, y devuelve true si se ha añadido y false en caso contrario.

A continuación, escribe una clase Pila, que implemente esta interfaz, utilizando para ello un array de Enteros y un contador de números almacenados.

Se deberá crear una excepción que tratará el problema de cuando intentemos extraer de la Pila un número o intentemos ver el primero que saldrá de la Pila en la situación en que la pila esté vacía.

Indicaciones para la implementación:

1. En la interfaz se declaran todos los métodos sin implementar ninguno.
2. Para la clase se utilizan como atributos un array de enteros y un entero que sirve de contador de números insertados.
3. El constructor recibe por parámetro el tamaño máximo de la pila.
4. El método **estaVacía()** comprueba si el contador es 0.
5. El método **añadir()** comprueba que cabe el elemento, y si es así, lo añade en la celda que indica el contador, posteriormente se incrementa el valor del contador. Devuelve true o false, según si se ha añadido o no.
6. **primero()** si está vacía, lanza PilaVacíaException, si no, devuelve el elemento que está indicado por el contador.
7. **extraer()** si está vacía, lanza PilaVacíaException, si no decrementa el contador y devuelve el elemento que está en la celda indicada por el contador, después de decrementarse.
8. Sobrecribir el método **toString()** (no estaría de más) para mostrar todos los número que contiene dentro.