

Práctica 1: Implementación en Java de Estructuras de Datos Lineales

El comportamiento del TAD Cola se puede caracterizar por la siguiente interfaz:

```
public interface Queue<E>{

    /**
     * Returns the number of elements in the queue.
     * @return number of elements in the queue.
     */
    int size();

    /**
     * Returns whether the queue is empty or not.
     * @return true if the queue is empty, false otherwise.
     */
    boolean isEmpty();

    /**
     * Inspects the element at the front of the queue.
     * @return element at the front of the queue.
     */
    E front();

    /**
     * Inserts an element at the rear of the queue.
     * @param element new element to be inserted.
     */
    void enqueue(E element);

    /**
     * Removes the element at the front of the queue.
     * @return element removed.
     */
    E dequeue();
}
```

Ejercicio 2: Diseñar y codificar la clase `ArrayQueue` (consultar las transparencias del primer tema) que permita construir colas basadas en arrays y que implemente la funcionalidad dada por la interfaz `Queue`. Usar excepciones (`RuntimeException`) en el caso de que alguna operación sea parcial. Finalmente, incluir la funcionalidad necesaria para que no se produzca un desborde del array.

Ejercicio 3: Diseñar y codificar la clase `LinkedListQueue` (consultar las transparencias del primer tema) que permita construir colas basadas en nodos enlazados y que implemente la funcionalidad dada por la interfaz `Queue`. La clase `Node` debe implementarse como una clase privada dentro de `LinkedListQueue`. Usar excepciones (`RuntimeException`) en el caso de que alguna operación sea parcial.