

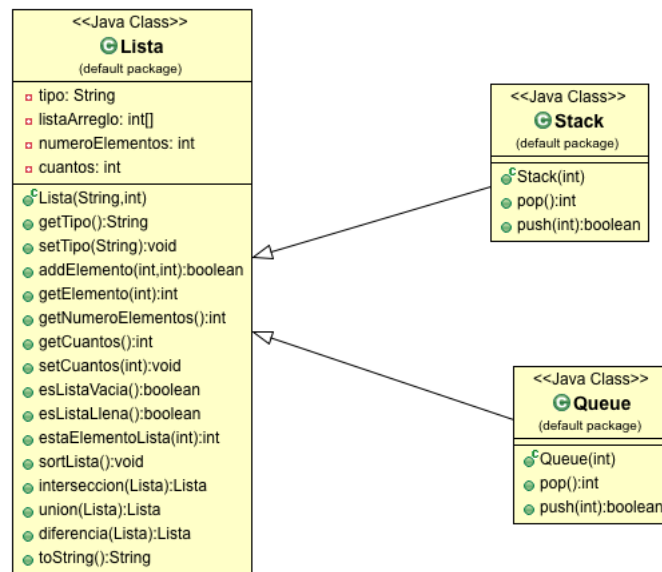


## PRACTICA 3 ESTRUCTURA DE DATOS Y POO

**Tema: Manejo de Arreglos y Herencia en Java**

**Practica: Individual**

Desarrollar las clases descritas en el diagrama de la figura 1.



**Descripción de Atributos de las clases:**

Clase	Atributo	Descripción
Lista	tipo	Guarda el string “Stack” o “Queue” que recibe del constructor de las clases hijas
	listaArreglo	Arreglo de enteros
	numeroElementos	Número de casillas que se abren en memoria para el arreglo. La da el usuario en la aplicación principal
	cuantos	Índice en el que se almacenan los elementos que se van pidiendo al usuario
Satck		
Queue		



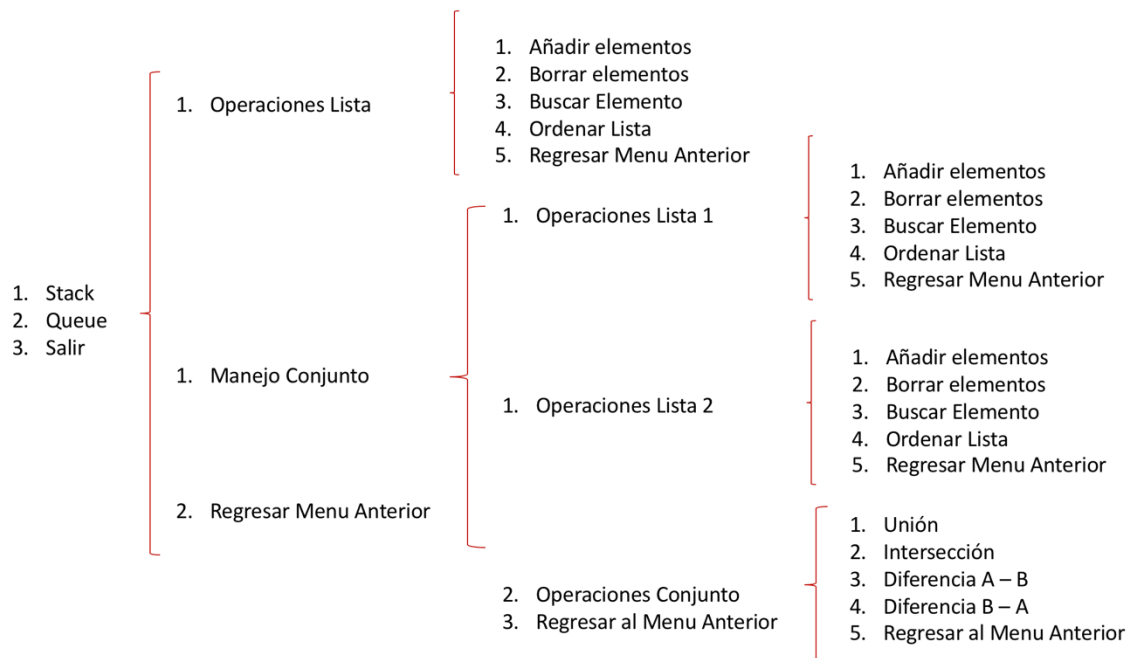
**Descripción de Métodos de las clases:**

<b>Clase</b>	<b>Método</b>	<b>Descripción</b>
Lista	addElemento	Recibe el elemento a añadir en la lista y el índice del arreglo en el que se va agregar ese elemento. Si existe elemento en la casilla indicada por el índice solo se sobreescribe la información. Regresa un booleano que será falso si el índice sale fuera del numero de casillas abiertas para el arreglo.
	getElemento	Recibe el el índice o localidad en la que el elemento se encuentra almacenado. Regresa el elemento en la casilla deseada. De no encontrar el elemento regresa un -1.
	esListaLlena	Método que regresa true si todas las localidades de memoria han sido utilizadas para guardar información.
	esListaVacía	Método que regresa true si todas las localidades de memoria han sido liberadas.
	estaElementoLista	Método que regresa la localidad del elemento que recibe como argumento. En caso de no encontrarlo regresa un -1.
	sortLista	Método que ordena de modo ascendente los elementos almacenados en la lista. (No debe ser método burbuja)
	union	Método que realiza la unión como operación de conjuntos con la lista que llega como argumento y el arreglo que tiene como atributo el objeto lista que invoque el método union. Regresa un objeto de tipo lista con el resultado de la operación.
	interseccion	Método que realiza la intersección como operación de conjuntos con la lista que llega como argumento y el arreglo que tiene como atributo el objeto lista que invoque el método interseccion. Regresa un objeto de tipo lista con el resultado de la operación.
	diferencia	Método que realiza la diferencia como operación de conjuntos con la lista que llega como argumento y el arreglo que tiene como atributo el objeto lista que invoque el método diferencia. Regresa un objeto de tipo lista con el resultado de la operación.
Stack	toString	Método que regresa un objeto de tipo String con los elementos de la lista. Ejemplo: "[2, 5, 7, 8, 9]"
	push	Método que añade el elemento que llega como argumento al inicio de la lista. Los elementos que se encuentran en la lista se van recorriendo hacia las localidades de mayor índice. Si la lista esta llena regresa un false. Si puede añadir el elemento regres un true.
	pop	Método que regresa el elemento almacenado en la primera casilla del arreglo. Los demas elementos deben ir subiendo de posición. En caso de no existir elemento que obtener de la primera posición regresa un -1.



Queue	push	Método que añade el elemento que llega como argumento en la lista. Los elementos subsecuentes deben ir llenando el resto de casillas libres. Si la lista esta llena regresa un false. Si puede añadir el elemento regres un true.
	pop	Método que regresa el elemento almacenado en la primera casilla del arreglo. Los demas elementos deben ir subiendo de posición. En caso de no existir elemento que obtener de la primera posición regresa un -1.

La aplicación principal debe presentar los siguientes menus:



1. La aplicación principal debe manejar un solo objeto de tipo Lista que será pasado como argumento a los diferentes métodos de la aplicación principal.
2. Cada menú se debe manejar en diferentes métodos en la aplicación principal.
3. En cada despliegue de menu debe aparecer al inicio el tipo de lista y sus elementos.
4. Al buscar elementos debe decir en que localidad se encuentra.
5. El tamaño del Stack o del Queue se determina en forma aleatoria.
6. Para las operaciones de conjunto se debe dar de alta una segunda lista a la que se le pueden aplicar las operaciones de lista.