

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 5 - Ciclo 2024</b>  <b>Tema: Listas</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

Este trabajo práctico permite conocer, distinguir, desarrollar y utilizar la o las implementaciones o estructuras de datos del tipo de dato abstracto List o Lista.

Este trabajo práctico también está pensado para que cada estudiante desarrolle habilidades de búsqueda y selección de información técnica necesaria para resolver los ejercicios.

#### **Importante:**

Cualquiera de los lenguajes de programación que se utilizan para realizar las prácticas seguramente facilita una clase genérica que permite crear objetos del tipo List<T>, sin embargo dicha clase no publica todos los métodos que se explican en teoría. En las clases teóricas de Estructura de Datos se desarrollan métodos particulares para que los estudiantes comprendan los diferentes mecanismos u operaciones que pueden realizarse con estructuras de datos del tipo List<T>; consecuentemente los estudiantes deberán codificar las implementaciones que se encuentran en las publicaciones de la asignatura.

#### **Ejercicios propuestos**

- 1) Implementar una clase “ListaEnlazadaSimple” que permita insertar, eliminar, buscar y mostrar elementos. Comprobar el correcto funcionamiento de la implementación propuesta creando una instancia de la lista. Se deben guardar X cantidad de objetos de la clase Empleado, la cual posee un legajo, dni, nombre, fecha de nacimiento y correo electrónico. Realizar las operaciones a pedido del usuario. Para ello, se deberá utilizar un menú de opciones.

##### Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 2) Crear una clase “Lista Doblemente Enlazada” con métodos para insertar elementos en cualquier posición. A partir de esa clase, crear una lista para guardar objetos de la clase Alumno, el cual tiene los siguientes atributos: lu, nombre, dni, fecha de nacimiento, correo electrónico y fecha de ingreso. El programa debe permitir que el usuario elija la posición en donde se inserta el nuevo alumno. Puede ser al principio de la lista, al final o en algún lugar del medio. En esta última opción se puede solicitar al usuario el ingreso de la posición.

##### Indicaciones:

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 5 - Ciclo 2024</b>  <b>Tema: Listas</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 3) Implementar métodos para eliminar elementos de una lista doblemente enlazada. La lista debe guardar objetos de la clase Deuda con atributos: codigo, descripcion, nombre del acreedor, fecha de la deuda y monto a pagar. El programa debe permitir eliminar una deuda buscada por código o por nombre de acreedor.

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 4) Crear un programa que permita combinar dos listas enlazadas ordenadas de números enteros en una sola, manteniendo el orden de los elementos.

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 5) Desarrollar métodos para recorrer una lista doblemente enlazada en ambas direcciones. El usuario debe elegir en qué sentido recorrer la lista. Al recorrer se deben mostrar los elementos de la lista sin eliminarlos.

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 6) Implementar un método que elimine todos los elementos duplicados de una lista enlazada, dejando elementos únicos. La lista de elementos puede ser simple o doblemente enlazada.

Estructura de Datos	<p align="center"><b>TRABAJO PRÁCTICO N° 5 - Ciclo 2024</b></p> <p align="center"><b>Tema: Listas</b></p>
<p align="center"><b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b></p> <p align="center"><b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b></p>	

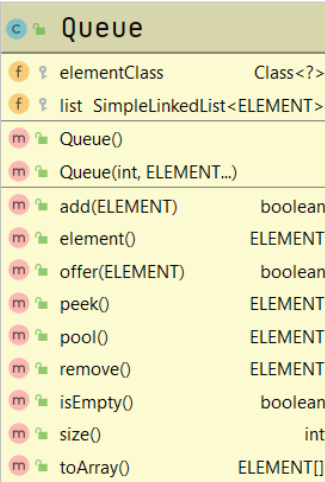
Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso** y **validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 7) Codificar una implementación de la clase Queue<ELEMENT> (Cola genérica) utilizando en la estructura interna una lista genérica. Usando dicha implementación, solicitar al usuario el ingreso de un número entero N, luego encolar N números aleatorios en la cola. Recorrer la cola mostrando los números ingresados y calcular el promedio de los mismos.

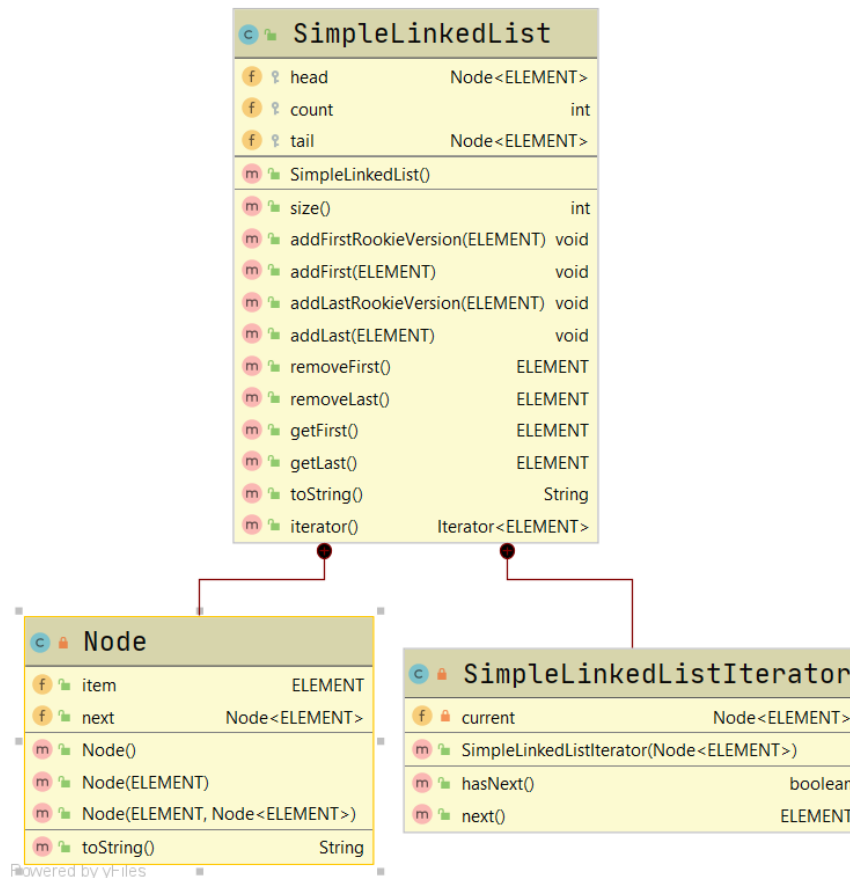
Para la implementación de la clase, tener en cuenta los diagramas presentados. En el primer diagrama se muestra el campo interno "list" de la clase Queue<ELEMENT> y en el segundo se presenta los detalles de la clase SimpleLinkedList<ELEMENT>.



Queue	
f	elementClass Class<?>
f	list SimpleLinkedList<ELEMENT>
m	Queue()
m	Queue(int, ELEMENT...)
m	add(ELEMENT) boolean
m	element() ELEMENT
m	offer(ELEMENT) boolean
m	peek() ELEMENT
m	poll() ELEMENT
m	remove() ELEMENT
m	isEmpty() boolean
m	size() int
m	toArray() ELEMENT[]

Powered by yFiles

Estructura de Datos	<p align="center"><b>TRABAJO PRÁCTICO N° 5 - Ciclo 2024</b></p> <p align="center"><b>Tema: Listas</b></p>
<p align="center"><b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b></p> <p align="center"><b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b></p>	



Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso** y **validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

El ejercicio debe implementar un mecanismo para seleccionar el ingreso de valores por consola o generados aleatoriamente.

- Codificar una implementación de la clase Stack<T> (Pila genérica) utilizando en la estructura interna una lista genérica. Usando dicha implementación, determinar si una palabra dada por el usuario es o no palíndromo. Una palabra es palíndromo cuando se lee igual de izquierda a derecha que de derecha a izquierda. Ejemplos de palíndromos: oso, somos, reconocer.

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 5 - Ciclo 2024</b>  <b>Tema: Listas</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso** y **validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

El ejercicio debe implementar un mecanismo para seleccionar el ingreso de valores por consola o generados aleatoriamente.