

## Actividad 2. Estructuras enlazadas

### Objetivo

Conocer cómo se gestionan las estructuras enlazadas (simple, doble y circular), mediante la implementación de un proyecto en java.

### Procedimiento

1. Leer los apuntes de Estructuras Enlazadas que están disponibles en **Faitic**, sección **Documentos e Ligazóns / Teoría / Estructuras enlazadas I**.
2. Emplear técnicas de aprendizaje colaborativo para resolver los ejercicios de clase que se indican en esta actividad, utilizando el lenguaje java. En concreto se utilizará la **técnica del rompecabezas**.
  - a. Se formarán grupos base de 3 personas (se formarán en clase de grupo reducido).
  - b. Cada miembro del grupo será responsable de la resolución de un ejercicio (ejercicios para resolver en clase)
  - c. En clase de grupo reducido se reunirán los expertos de cada ejercicio para resolverlo en común.
  - d. Después cada persona volverá a su grupo base para explicarles a sus compañeros lo que ha aprendido.
3. Los *otros ejercicios propuestos* en esta actividad se trabajarán de manera presencial/no presencial.

### Evaluación

El trabajo colaborativo será evaluado de manera grupal el día 19 de febrero de 2020.

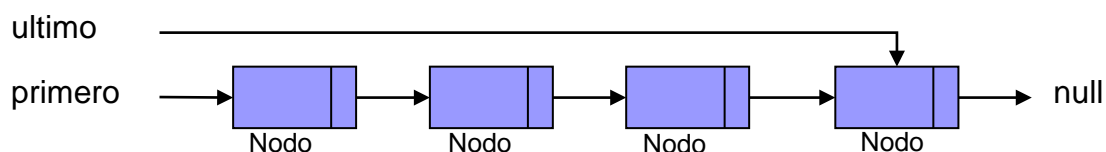
Estos contenidos serán evaluados mediante una prueba individual el 11 de marzo de 2020.

### Tiempo estimado

4 horas de trabajo no presencial + 4 horas de trabajo presencial.

### Ejercicios para resolver en clase

**1.-** Se quiere implementar una lista enlazada de modo simple que almacena números enteros, con referencias al primer y último nodo de la estructura.



Se pide escribir un proyecto en java que implemente:

- a.- la clase *Nodo* con sus atributos, método constructor y métodos observadores y modificadores.

b.- la clase *EnlazadaSimple2Refer*, con sus atributos y los métodos siguientes:

Crear la lista vacía, es decir, sin nodos (método constructor)

Contar el número de elementos de la lista

Contar el número de veces que aparece un elemento en la lista

Saber si un elemento que se pasa como parámetro está en la lista

Insertar un elemento al principio de la lista

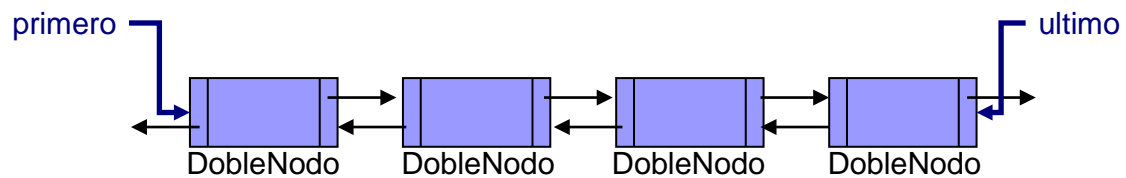
Insertar un elemento al final de la lista

Borrar un elemento que se pasa como parámetro

Método toString()

c.- Crear una clase Prueba que cree una lista de enteros y que contenga sentencias que invoquen a los métodos implementados en el apartado anterior.

2.- Se quiere implementar una lista doblemente enlazada de números enteros con referencias al primer y último nodo de la estructura.



Se pide escribir un proyecto en java que implemente:

a.- La clase *DobleNodo*, con sus atributos y métodos observadores y modificadores.

b.- La clase *DoblementeEnlazada*, con sus atributos y los métodos siguientes:

Crear la lista vacía, es decir, sin nodos (método constructor)

Contar el número de veces que aparece un elemento en la lista

Insertar un elemento al principio de la lista

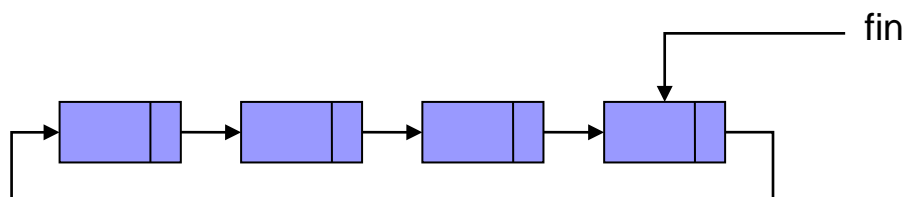
Insertar un elemento al final de la lista

Borrar un elemento que se pasa como parámetro

Método toString()

c.- Crear una clase Prueba que implemente un método main encargado de crear una lista de enteros y de invocar a los métodos implementados en el apartado anterior.

3.- Se quiere implementar una lista circular enlazada de números enteros y con referencia al último nodo de la estructura.

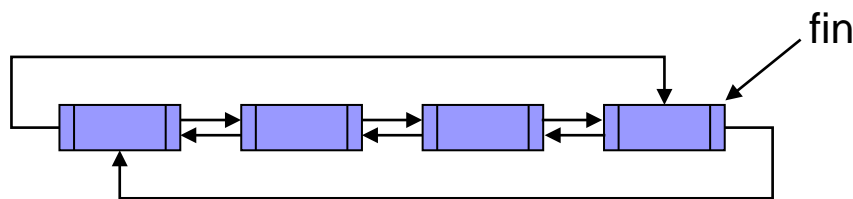


Se pide escribir un proyecto en java que implemente:

- a.- La clase *Nodo*, con sus atributos y métodos observadores y modificadores.
- b.- La clase *EnlazadaCircular*, con sus atributos y los métodos siguientes:
  - Crear la estructura vacía (método constructor)
  - Contar el número de veces que aparece un elemento en la lista
  - Devolver el elemento que está de primero en la lista; si la lista está vacía devuelve -0.
  - Insertar un elemento al final de la lista
  - Borrar el elemento al principio de la lista
  - Método toString()
- c.- Crear una clase Prueba que cree una lista de enteros y que contenga sentencias que invoquen a los métodos implementados en el apartado anterior.

## Otros ejercicios propuestos

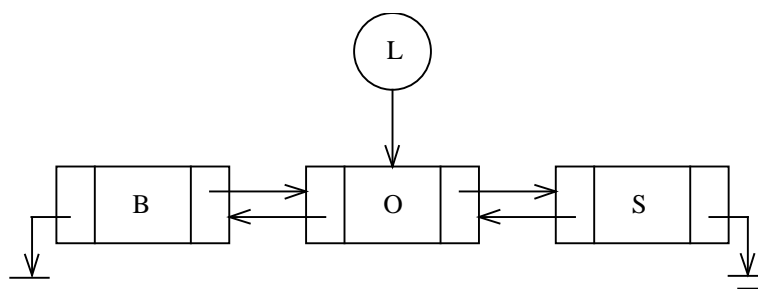
- 4.- Se quiere implementar una lista circular doblemente enlazada de números enteros y con referencia al último nodo de la estructura.



Se pide escribir un proyecto en java que implemente:

- a.- La clase *DobleNodo*, con sus atributos y métodos observadores y modificadores.
- b.- La clase *DoblementeEnlazadaCircular*, con sus atributos y los métodos siguientes:
  - Crear la estructura vacía (método constructor)
  - Contar el número de veces que aparece un elemento en la lista
  - Devolver el elemento que está de primero en la lista; si la estructura está vacía, devuelve null
  - Insertar un elemento al final de la lista
  - Borrar el elemento al principio de la lista
  - Método toString()
- c.- Crear una clase Prueba que cree una lista de enteros y que contenga sentencias que invoquen a los métodos implementados en el apartado anterior.

- 5.- Supongamos la siguiente estructura para almacenamiento de una secuencia de elementos clasificados en orden ascendente. Cada elemento está enlazado con su anterior y su siguiente. Se mantiene, además, un puntero a la posición central de la secuencia. ¿Qué ventajas e inconvenientes tiene esta estructura? Define los tipos de datos más adecuados para soportar esta estructura. Implementa una operación que permita insertar un elemento. Las inserciones se realizan manteniendo el orden de los elementos y el apuntador al elemento central, si es posible.



**6.-** Tenemos una clase Alumno que almacena el DNI, nombre, número de matrícula y curso de un alumno de la ESEI. Interesa mantener ordenados dichos alumnos tanto por nombre como por número de matrícula. La solución escogida ha sido utilizar una lista enlazada donde cada nodo tiene dos campos de enlace: uno para ordenar ascendente los alumnos según su nombre y otro para ordenar descendente los alumnos según su número de matrícula. Define las clases necesarias y escribe un método que *inserte* un nuevo elemento en la estructura.

