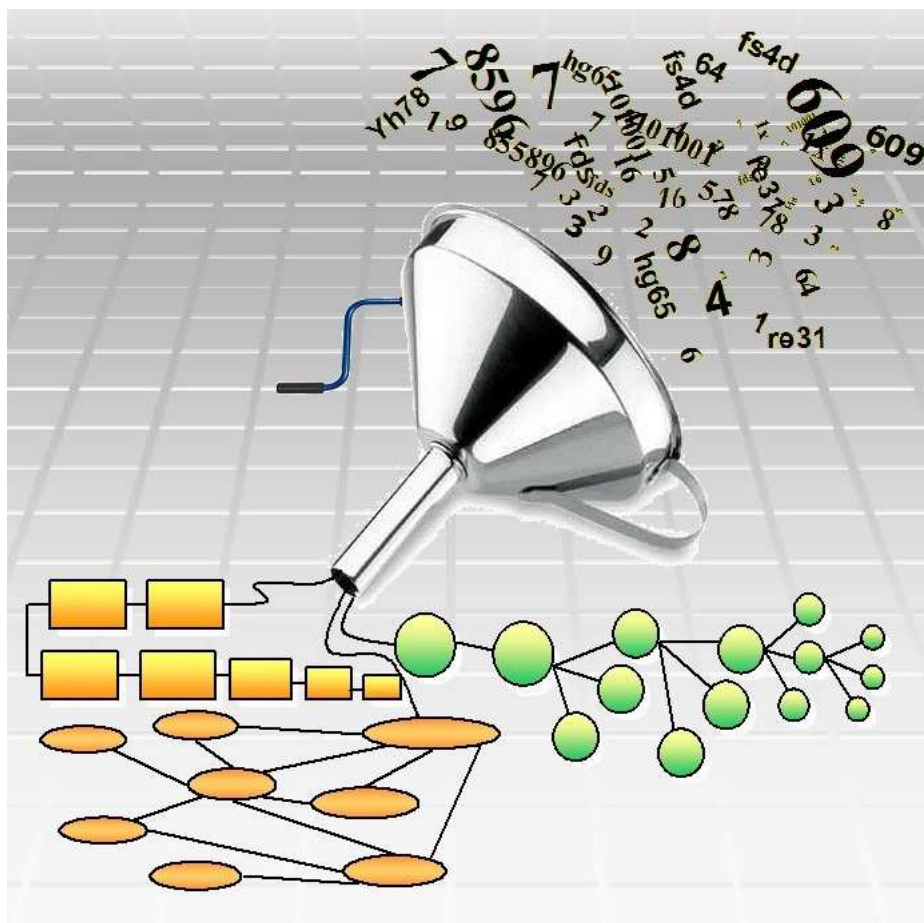


# Memoria Practica 01



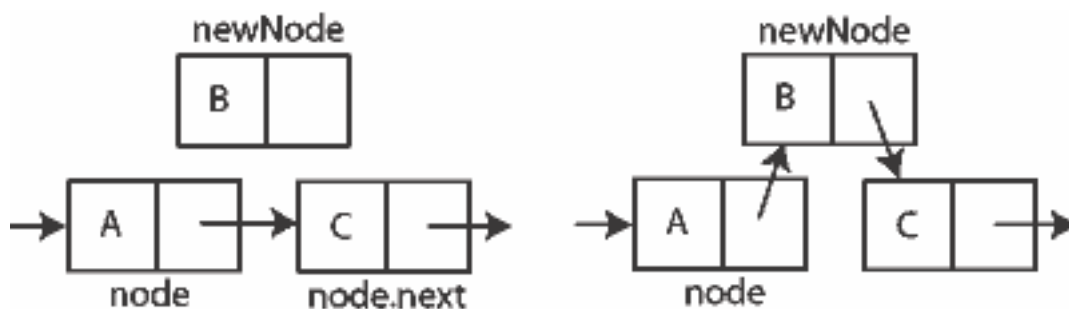
<b><u>Autor:</u></b>	David Subires Parra
<b><u>Asignatura:</u></b>	Estructuras de Datos y Algoritmos 1
<b><u>Grado:</u></b>	Ingeniería informática
<b><u>Universidad:</u></b>	Universidad de Almería
<b><u>Curso Académico:</u></b>	2013/2014

## Ejercicio 01

- De las principales colecciones lineales implementados en la JCF de Java (`ArrayList`, `LinkedList`, `Vector`, `Stack`, etc), razone detalladamente cuál es la que cree más conveniente para resolver este ejercicio.

Creo que la colección lineal más conveniente para este ejercicio es la colección usada, `LinkedList`, ya que la ventaja de esta sobre las demás es la eficiencia en la operación añadir elementos entre dos elementos, es decir, añadir elementos en cualquier posición que no sea la primera ni la última. Operación que se realizará a menudo en este ejercicio.

Esto es así porque al añadir un elemento entre otros dos elementos, en una lista enlazada, la operación interna que realiza es “desenganchar los dos elementos” para unir entre medias el nuevo.



- Para la resolución de este ejercicio se ha propuesto utilizar el `LinkedList`, ¿sería muy complicado realizar la misma implementación con `ArrayList`?, ¿qué habría que hacer?.

Sería muy sencillo, lo lograríamos solo con dos modificaciones en el código.

Primero donde se declara el atributo `text`

`Protected LinkedList<String> text` → `protected ArrayList<String> text;`

Y por ultimo en el único constructor, cambiar

`text = new LinkedList<String>()` → `text = new ArrayList<String>;`

## Ejercicio 02

- Para finalizar, como ya hemos podido comprobar, para la resolución de este ejercicio se ha propuesto utilizar el `ArrayList`. ¿Sería apropiado realizar la misma implementación con `LinkedList`?, ¿qué habría que hacer?. Contestar a estas preguntas en el archivo PDF correspondiente a la práctica 1 (`practica_01.pdf`), separando convenientemente las respuestas de los diferentes ejercicios que componen la práctica.

No sería apropiado implementar este ejercicio con un `LinkedList`, ya que en este caso, la operación más usual sería recorrer los datos, para la cual `ArrayList` nos ofrece mayor eficiencia.

Para realizar la implementación con `LinkedList`, una vez más, solo tendríamos que cambiar la línea donde se declara el atributo `ArrayList<ProyectoCiudades>` y también la inicialización en los dos constructores. Reemplazando `ArrayList` por `LinkedList`.

Además de cambiar el tipo de objeto que devuelven los métodos `getCiudades` y `getProyectoCiudades`.