



ESCOLA TÈCNICA SUPERIOR  
D'ENGINYERIA  
Universitat Rovira i Virgili



# Estructura de dades

## Pràctica 1. Llista doblement encadenada i Skip List

curs 2020-21

Estudiant: Alexandru Chitenco

Professor/a: Clara Granell

Data de lliurament: 17/06/2021

## 1. Introducció

Para esta práctica debemos estudiar e implementar un SkipList y una Lista Doblemente Encadenada.

## 3. Descripció dels requeriments

Los requisitos de nuestro software son:

1. Crear dos tipos de listas distintas para luego comparar su coste computacional.
2. Crear una interfaz comuna para las dos la cual tendrá los métodos comunes requeridos en el enunciado.
3. Las dos listas deberán usar de elementos un genérico y ser comparables genéricos.
4. La lista doblemente encadenada adicionalmente deberá de implementar Iterable genérico.

## 4. Decisions de disseny

*Lista doblemente encadenada:*

Para la lista doblemente encadenada las decisiones de diseño más importantes han sido la de en qué consistirá el nodo. El nodo diseñado tiene básicamente tres atributos con sus respectivos getters y setters, el valor, el nodo siguiente de la lista y el nodo anterior.

Para la creación de la lista en sí cree otra clase que la llame IteratorList para poder cumplir con los requisitos del enunciado e implementar las funcionalidades Iterator en mi lista. En general la creación de la lista no ha llevado gran complicación ya que es un tipo de estructura bastante sencilla.

*SkipList:*

Para la lista SkipList el nodo exige más complicación debido a luego las buenas funcionalidades que aporta la misma, por lo tanto decidí implementar dos atributos, el valor y un array de nodos de longitud dependiente de los niveles que tenga el nodo, que eso se decide dentro de la creación de la lista.

En cuanto a niveles decidí poner como máximo 10 para poder optimizar adecuadamente los resultados de costes computacionales.

En cuanto a la lista en sí, lo más complicado fue insertar un nodo. En la inserción se decide los niveles que tiene cada nodo de manera aleatoria.

## Conclusions

Para concluir la práctica he de decir que a gran escala la SkipList ha resultado ser muchísimo más eficiente en la búsqueda(en el análisis de costes computacionales se puede observar mejor). Aunque también es cierto que la formación de la lista tiene muchísimo más coste computacional.

## Observaciones

La práctica la realicé en IntelliJ y se puede observar el progreso en mi GitHub

<https://github.com/liteJoseStalin/LinkedListAndSkipList>