

## 2° PARCIAL DE ALGORITMICA Y ESTRUCTURA DE DATOS II - 2018

1) Incluir en el TAD **List** el método *RemoveAll* y realizar su implementación en la clase **ArrayList**. Tener en cuenta que la lista puede tener elementos duplicados, si los hay remover todas sus ocurrencias. Realizar una aplicación que pruebe el mismo.

```
/*
 * Remueve de la lista todos los elementos que están contenidos en la
 * colección especificada.
 * Retorna true si la lista cambia como resultado de su llamada.
 */
public boolean removeAll(Iterable<E> i)
```

2) Realizar implementaciones en la clase **LinkedBinaryTree** para verificar si un árbol binario está lleno y si dos árboles binarios son semejantes (considere que los árboles no tienen elementos duplicados). Realizar una aplicación que pruebe el mismo.

a) **Arbol binario lleno**: Es aquel árbol en el que los nodos de cada nivel tienen sus dos hijos o ninguno (si es hoja).

```
public boolean lleno()
```

b) **Arbol binario semejante**: Dos árboles binarios son semejantes si tienen el mismo número de nodos y los valores de los nodos del primer árbol son los mismos valores de los nodos del segundo árbol, sin importar la relación de parentesco entre ellos.

```
public boolean semejante(LinkedBinaryTree<E> t)
```

3) Desarrollar una aplicación que cargue en un árbol binario ordenado una serie de palabras en español y su correspondiente traducción al inglés. Recorrer el mismo y generar un nuevo diccionario que dada una palabra en inglés retorne su correspondiente traducción.

---

### TEORIA (Entregar en una hoja aparte)

- 1) Defina que es un iterador. Explique cuál es su función, como se implementa en Java y qué características tienen ambos.
- 2) TAD.
  - a. Definición
  - b. En el TAD árbol que significa el concepto Posición. ¿Qué métodos soporta?
  - c. Qué tipos de recorrido de árboles conoce y de un ejemplo de uso de cada uno
  - d. Defina árbol binario, árbol binario de búsqueda y dé un ejemplo de uso de este último.