

Todas las preguntas tienen la misma ponderación (25%).

1. Suponer que tenemos definidos los TAD **Natural** y **Booleano**. Queremos definir el TAD genérico **Lista[T]**, para almacenar listas de elementos de tipo **T**. El tipo **T** debe tener definida una operación de comparación entre sus elementos, que tendrá el nombre **esMenor(*t1*, *t2*)**, que indica si *t1* es menor que *t2*; y una operación **esInteresante(*t*)**, que indica si el elemento *t* es interesante o no. Construir una especificación formal, usando el método axiomático, del TAD **Lista[T]**. El tipo deberá tener las siguientes operaciones:
 - **vacía**: crea una nueva lista vacía.
 - **insertar(*elemento*, *lista*)**: inserta un nuevo elemento en una lista.
 - **invertir(*lista*)**: devuelve una lista con los mismos elementos, pero invirtiendo el orden de los elementos.
 - **interesantes(*lista*)**: devuelve una lista con los elementos de *lista* que sean interesantes (de acuerdo con la operación **esInteresante**), en el mismo orden que la lista original.
 - **contarMenoresInteresantes(*valor*, *lista*)**: dada una lista y un *valor* de tipo **T**, devuelve un natural que indica el número de elementos interesantes de la lista que son menores que *valor*.

Escribir las cuatro partes de la especificación axiomática (nombre, conjuntos, sintaxis y semántica) del tipo genérico **Lista[T]**.

2. En una aplicación necesitamos almacenar información sobre los productos que vende una tienda de informática. Cada producto tiene un identificador, un nombre y una descripción en forma de texto; los dos últimos campos son cadenas de texto, que pueden dividirse en palabras, mientras que el identificador es un número de 20 dígitos. Queremos desarrollar una aplicación que nos permita encontrar rápidamente un producto. La búsqueda puede ser por el identificador o por palabras (tanto del nombre como de la descripción del producto, de manera indiferente).

Describir cómo se utilizarían las tablas de dispersión para resolver este problema, optimizando las búsquedas por los dos criterios indicados. Se debe indicar la tabla o tablas de dispersión que se usarían, proponer funciones de dispersión y de redispersión adecuadas, qué datos se almacenan, qué tamaños de tablas usar, cómo hacer las consultas, etc. Hacer una estimación de la eficiencia conseguida para las operaciones de consulta.

3. Contestar las tres siguientes preguntas justificando las respuestas.
 - a) En un árbol AVL vacío se insertan los elementos 73, 82, 26, 37, 19, 90, 32, 44, 88, 50, 1. Después se eliminan 37, 73, 50, 9. Mostrar los casos que se producen y, en caso necesario, las rotaciones que se aplican.
 - b) Suponer que los mismos datos se almacenan en un árbol B de orden 5. Mostrar las inserciones y eliminaciones de los elementos.
 - c) Hacer una estimación del uso de memoria tanto del árbol AVL como del árbol B anteriores, después de hacer las inserciones. Suponer que en ambos casos se almacenan pares (clave= entero; valor= registro de 100 bytes).
4. Escribir un algoritmo eficiente que dado un grafo dirigido sin pesos encuentre para cada nodo del grafo el ciclo más corto que pasa por ese nodo. Suponer que el grafo tiene **n** nodos, numerados desde 1 hasta **n**, y que podemos acceder a los adyacentes de un nodo con la operación: para cada nodo *w* adyacente a *v* hacer.