Ismael Lautaro Martner Varela

1. Explica la diferencia conceptual entre los tipos Nodo<E> y E.
El tipo nodo es basicamente un wrapper de E que contiene la direccion del nodo siguiente y anterior. La funcion del nodo es basicamente estructural a diferencia de E que cumple una funcion semantica al ser realmente lo que nos interesa de la lista. Pero E puede tambien ser un nodo entonces realmente no hay una diferencia muy profunda entre ellos fuera de su funcion en las listas.

2. ¿Por qué ListIterator sólo permite remover, agregar o cambiar datos después de llamar previous o next?
Porque si no implica que no se ha inicializado el iterador, i.e. el iterador aun no existe. No puedes aplicar operaciones a algo que no existe.

3. Si mantenemos los elementos ordenados alfabéticamente, por ejemplo, ¿cuándo sería más eficiente agregar un elemento desde el inicio o el final de la lista?
Ninguna opcion es mas eficiente que la otra, lo mas eficiente seria agregarlos desde la mitad.
No estoy seguro a que se refiere la pregunta, ojala la haya entendido bien.

4. En qué casos sería más eficiente obtener un elemento desde el inicio de la lista o desde el final de la lista.
(Pregunta ambigua)
Cuando el elemento esta mas cerca del principio que del final seria mas eficiente obtenerlo desde el inicio de la lista. Si esta mas cerca del final que del inicio seria mas eficiente obtenerlo desde el final.