

Ventajas e inconvenientes de las implementaciones estática y dinámica del TAD Lista.

	Estática	Simple Enlace	Doble Enlace
Necesidad de memoria	mucha	menos en promedio (más que simple enlace)	
Memoria contigua ¹	si	no	
Acceso directo ²	si	no	
Ampliable	no	si	
Operaciones más costosas ³	insertarDato eliminarPosicion (excepto al final)	insertarDato (final) eliminarPosicion (final) anterior, ultima borrarLista, copiarLista	insertarDato (final) ultima borrarLista copiarLista
Seguridad ⁴	Total	Relajada	

¹ En la realización con estructuras estáticas tenemos espacio de memoria fijo, por lo que, en general, utilizamos más memoria de la que necesitamos. Por otro lado, en las estructuras dinámicas cada nodo ocupará un espacio de memoria adicional correspondiente al puntero (a igualdad de elementos ocuparán más).

² Con la estática tenemos acceso directo a los elementos de la lista.

³ En general, en las estructuras estáticas son más sencillas las operaciones que requieren acceder a los elementos de la lista por su posición. Por el contrario, hemos de realizar desplazamiento de los elementos cuando queremos insertar o eliminar uno de ellos.

Las operaciones no mencionadas en este apartado tienen igual complejidad en todas las implementaciones (ej. buscarDato).

⁴ Cuando manejamos arrays, para alterar la lista hay que pasar la variable de tipo `tlista` por referencia. Sin embargo esto no ocurre así con los punteros, ya que hemos visto que pasar un puntero como parámetro equivale a pasar por referencia la variable a la que apunta (véase, p.e., la implementación de `ActualizarDato`).

La implementación concreta que elijamos dependerá de las características que deba tener la lista para nuestra aplicación y de las operaciones que vayamos a realizar. P.e., si el tamaño es más o menos conocido, si va a tener pocos cambios una vez establecida y se suelen hacer muchos accesos por posición, lo más apropiado, probablemente sea una implementación basada en arrays. Por otro lado, si la lista va a ser muy dinámica en cuanto a su tamaño o con frecuentes inserciones y eliminaciones de elementos, lo más apropiado será una implementación basada en variables dinámicas.