

Estructuras de Datos

Práctica 3. Listas enlazadas

Sesiones de prácticas: 2

Objetivos

Construir estructuras de datos formadas por nodos enlazados

Ejercicio propuesto

Aunque el uso de vectores en la práctica 2 mejora mucho la implementación de la práctica 1, el borrado que se realiza en posiciones intermedias sigue siendo ineficiente. En esta práctica proponemos sustituir el vector dinámico por una lista doblemente enlazada. Como siempre la primera parte de la práctica consistirá en la implementación de la estructura de datos como una plantilla genérica y en la segunda parte la aplicaremos a nuestro problema.

Las operaciones a implementar en la lista doblemente enlazada (*Lista<T>*) son:

- Constructor por defecto, constructor copia, operador de asignación y destructor.
- Operación *tam()* para obtener el número de elementos en la lista.
- Operaciones *Iterador<T> iteradorIni()* y *Iterador<T> iteradorFin()* para obtener iteradores apuntando al principio y final respectivamente. Previamente se habrá definido la plantilla *Iterador<T>* correspondiente.
- *void insertar(Iterador<T> &it, const T &dato, void borrar(Iterador<T> &it), insertarFin(const T &dato)* y *void borrarFin()* para insertar o borrar un dato en la posición indicada por el iterador (incluyendo el principio), o insertar o borrar al final.

Ejercicio voluntario

En nuestra simulación vamos a incluir una nueva suposición: las partículas con una masa mayor a 1 son inestables y tienen una probabilidad de desintegración directamente proporcional a la masa (*m*) de ésta, siguiendo la fórmula:

$$P_{desint.}(m) = \begin{cases} 0 & \text{si } m = 1 \\ 1 - e^{-0,2m} & \text{si } m > 1 \end{cases}$$

La desintegración implicará la división de la partícula en dos partículas de mitad de tamaño aproximadamente. Una de las partículas ocupará la posición inicial de la partícula dividida y la otra se situará en una posición adyacente libre. Realizar los cambios necesarios en el algoritmo para implementar esta nueva funcionalidad.