

Estructuras de Datos

Práctica 2. Vectores dinámicos

Sesiones de prácticas: 2

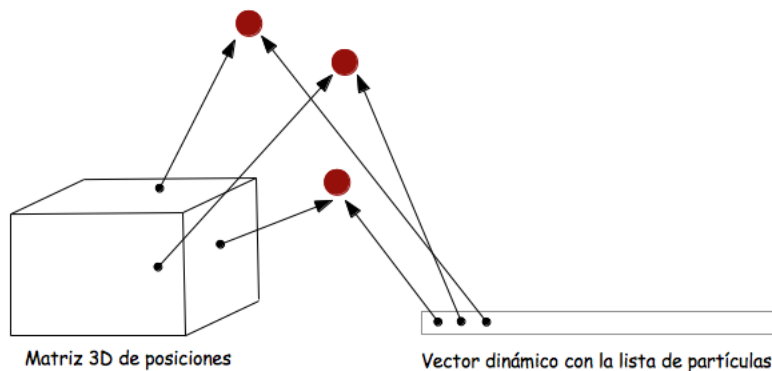
Objetivos

Construir estructuras dinámicas en memoria

Ejercicio propuesto

La implementación realizada en la práctica 1 es extremadamente ineficiente porque requiere revisar todas las posiciones del espacio (100x100x100) en cada iteración para localizar las partículas y actualizar su posición. Vamos a añadir un vector donde tendremos la lista de punteros a las partículas existentes en el mundo. El funcionamiento de esta nueva implementación será el siguiente:

1. Iniciamos la matriz y añadimos al vector los punteros a las partículas
2. Iteramos sobre las partículas usando el vector. Por cada partícula:
 - a) Calculamos nueva posición.
 - b) Si la nueva posición está libre, movemos la partícula actualizando la posición en la matriz.
 - c) Si en la nueva posición hay una partícula, fusionamos y eliminamos la partícula de la matriz y del vector.



Para esta nueva versión implementaremos como paso previo un vector dinámico (clase `VectorDin<T>`) con las siguientes operaciones:

- Un constructor por defecto que creará el vector con tamaño físico inicial 1 y lógico 0.
- Constructor copia, operador de asignación y destructor correspondiente.
- `tam()`, que devolverá el tamaño lógico del vector.
- El operador `[]` para devolver el contenido de la posición indicada por referencia para que sea modificable.
- `insertarFinal(...)` para añadir un nuevo dato en la última posición.
- `insertar(...)` para añadir un dato en una posición arbitraria del vector.
- `borrarFinal(...)` para eliminar el último dato del vector.
- `borrar(...)` para eliminar un dato en una posición arbitraria del vector.