

Tarea 21

1 Contenedores e iteradores

Ejercicio 1 *A la manera del ejemplo de la clase, implementar un programa usando un `set<int>` que lea un archivo de texto y almacene en esa estructura todos los caracteres del archivo. Pedir al usuario que entre un carácter y buscar si este carácter esta presente o no en el archivo.*

***Ejercicio 2** *Documentarse en como está implementado en la librería estándar el SET: ¿cómo guarda los datos ordenados? ¿cómo se asegura que en el peor caso la búsqueda de un elemento no se haga con una complejidad importante? Intentar aislar el tipo exacto de la estructura de datos utilizada para la implementación.*

Ejercicio 3 *Comprobar el comportamiento del vector cuando llega a saturación de su capacidad: mostrar (usando por ejemplo objetos que imprimen mensajes al llamar constructor y destructor) que todos los objetos presentes están copiados en nuevos objetos y destruidos, mostrar también que la política usada para manejar las redimensionalización sí consiste en doblar la capacidad en cada saturación.*

Ejercicio 4 *Implementar una clase patron `myDeque` con un **arreglo circular** que tenga:*

- *un iterator simple “iterator” (implementado como simple apuntador),*
- *un constructor por default,*
- *un constructor por tamaño inicial,*
- *un constructor por numero de elementos y una referencia a un objeto que duplicar este numero de veces,*
- *metodos `back()`, `front()`, `begin()`, `end()`, `size()`, `clear()`,*
- *un operador `[]`,*
- *métodos `push_back()`, `push_front()`, `pop_back()`, `pop_front()`*

Este arreglo tendrá una capacidad limitada, y en caso de saturación se emitirá una excepción si el usuario quiere agregar un elemento más.

Ejercicio 5 *Por default, la pila stack de la librería estándar está implementada con un deque. Escribir un ejemplo en qué se usará una versión de la stack implementada con un vector.*

Ejercicio 6 *Por default, la cola de prioridad `priority_queue` de la librería estándar está implementada con un vector. ¿Como funcionará? Es decir, ¿como los datos estarán guardados en memoria en este vector, de tal manera a que la operación `top()` sea “rápida”?*