



Universidad  
de Jaén

Departamento de Informática

## Prácticas de Estructuras de Datos

Grado en Ingeniería en Informática

Curso 2021/2022

# Práctica 1. Implementación de vector dinámico mediante plantillas y operadores en C++

## Sesiones de prácticas: 2

### Objetivos

Implementar la clase `VDinamico<T>` utilizando **patrones de clase y excepciones**. Programa de prueba para comprobar su correcto funcionamiento.

### Descripción de la EEDD

Implementar la clase `VDinamico<T>` para que tenga toda la funcionalidad del vector dinámico descrita en la Lección 4, utilizando patrones de clase y excepciones. Los métodos a implementar serán los siguientes:

- Constructor por defecto `VDinamico<T>`, iniciando el tamaño físico a 1 y el lógico a 0.
- Constructor dando un tamaño lógico inicial, `VDinamico<T>(unsigned int tamlog)`, iniciando el tamaño físico a la potencia de 2 inmediatamente superior a tamlog (tamaño lógico).
- Constructor copia `VDinamico<T>(const VDinamico<T>& origen)`.
- Constructor de copia parcial `VDinamico<T>(const VDinamico<T>& origen, unsigned int posicionInicial, unsigned int numElementos)`. En este constructor hay que tener en cuenta que el vector que se genera debe tener un tamaño físico potencia de 2.
- Operador de asignación (`=`).
- Operador `[]` para acceder a un dato para lectura/escritura.
- Insertar un dato en una posición: `void insertar(const T& dato, unsigned int pos = UINT_MAX)`. Si no se indica la posición (valor `UINT_MAX`) entonces la inserción se realiza al final del vector. Esta operación siempre incrementa el tamaño lógico en 1.
- Eliminar un dato de una posición intermedia en  $O(n)$ : `T borrar(unsigned int pos = UINT_MAX)`. Si no se indica la posición (valor `UINT_MAX`) entonces se elimina el último dato del vector.
- Ordenar el vector de menor a mayor. Se puede utilizar la función `sort` de `<algorithm>`: `void ordenar()`.
- Ordenar el vector de mayor a menor con la función `void ordenarRev()`.

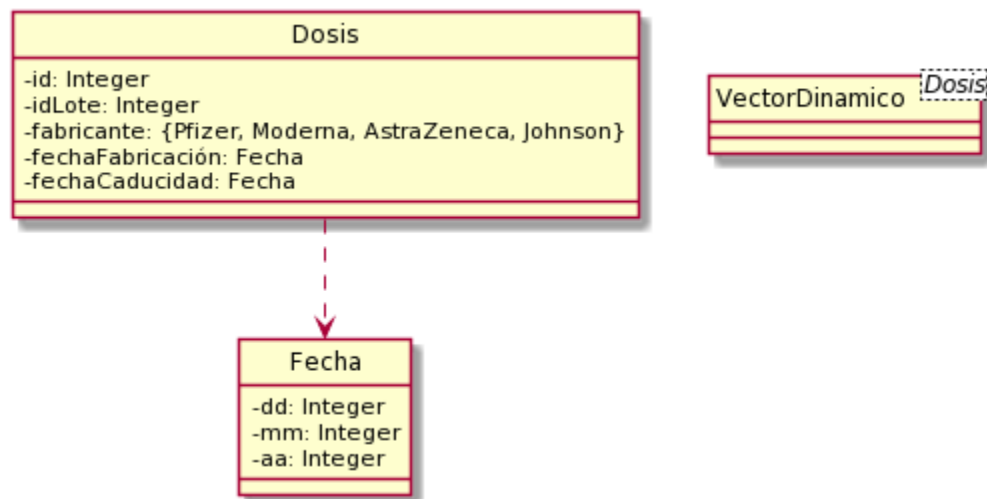
- Buscar un dato en el vector utilizando el método de búsqueda binaria o dicotómica y devolviendo la posición del dato. `int busquedaBin(T& dato)`. Se crea un objeto básico T que contenga el atributo objeto de la búsqueda que se rellena si la búsqueda tiene éxito. Sino devuelve -1.
- `unsigned int tamlog()` para obtener el tamaño (lógico) del vector.
- El destructor correspondiente.

### Programa de prueba: Creación de un vector dinámico de dosis de vacunas

Este curso vamos a realizar una aplicación para la gestión de la vacunación de COVID-19. Para ello trabajaremos con clases de objetos relacionados con la temática, una de ellas las dosis de las vacunas.

En esta primera práctica almacenaremos los datos del fichero adjunto en un vector dinámico, es decir, el vector se instanciará con los datos de tipo Dosis. Para ello se deberá leer el fichero adjunto que almacena la información de una dosis por línea (fabricante es de tipo enumerado). Los valores de los atributos están puestos en orden según el siguiente UML. El código para la lectura de dicho fichero y la clase Fecha se encuentran adjuntos a la práctica.

La clase Dosis contendrá únicamente los constructores, operadores y getter y setter necesarios para la práctica. El criterio de ordenación de Dosis es por id.



La prueba implementada en la función `main ()` consistirá en:

- Después de implementar el vector y la clase Dosis, ejecutar el programa de prueba adjunto y comprobar que funciona correctamente. Cargando un total de 10000 dosis de vacuna.
- Instanciar el vector con todos los objetos de tipo Dosis contenidos en el fichero adjunto, la fecha de caducidad se debe calcular como dos meses a partir de la fecha de fabricación. Mostrar los identificadores de las primeras 50 dosis.

- Ordenar el vector al revés, es decir, de mayor a menor y mostrar los identificadores de las primeras 50 dosis.
- Ordenar el vector de menor a mayor y mostrar los identificadores de las primeras 50 dosis.
- Una vez ordenado el vector, buscar las dosis con los identificadores 346335905, 999930245, 165837, 486415569 y 61385551, mostrando su posición en el vector. Teniendo en cuenta que pueden no existir.
- Se ha detectado un problema en el sistema de control y es necesario aislar dosis. Crear un nuevo vector del tipo VDinamico<Dosis> que contenga todas las dosis defectuosas. Esas dosis están entre los ids 10000 y 2500000, su id de lote es múltiplo de 2 o de 5 y se fabricaron en el 2020. Borrar las dosis defectuosas del vector inicial, mostrar el tamaño lógico de ambos vectores y toda la información de las primeras 5 dosis defectuosas. Si se usan vectores auxiliares, deberán declararse como VDinamico.
- Aquellos que hagáis las prácticas por parejas tendréis que estudiar los tiempos de los cuatro apartados anteriores. Para el resto es opcional.

### **Estilo y requerimientos del código:**

1. El código debe ser claro, tener un estilo definido y estar perfectamente indentado, para ello se pueden seguir algunos de los estilos preestablecidos para el lenguaje C++ (<http://geosoft.no/development/cppstyle.html> ).
2. Deben comprobarse todas los posibles errores y situaciones de riesgo que puedan ocurrir (desbordamientos de memoria, parámetros con valores no válidos, etc.) y lanzar las excepciones correspondientes, siempre que tenga sentido. Leer el tutorial de excepciones disponible en el repositorio de la asignatura en docencia virtual.
3. Se valorará positivamente la calidad general del código: claridad, estilo, ausencia de redundancias, etc.