PRÁCTICA 4 **Estructuras de Datos y Algoritmos**

Título: Programación de TAD genérico arborescente en C++. **Curso 2025/26**

Objetivo:

- Diseñar una implementación arborescente y en memoria dinámica en C++ del TAD genérico *"colecciones de elementos interdependientes"*, particularizarlo y utilizarlo con un TAD no genérico *evento*, para implementar un programa que los utilice para gestionar una colección de eventos que puedan detectarse en un sistema de seguridad.

Fecha límite de entrega: **23-11-2025 (incluido)**

Descripción detallada:

Se trata de replicar lo realizado en la Práctica 3, pero desarrollando una **implementación dinámica utilizando un árbol binario de búsqueda** del TAD genérico *colecInterdep*. Al igual que en la Práctica 3, los TADs implementados deberán corresponder a las especificaciones dadas en el enunciado de la Práctica 3 (los TAD *evento* y *colecInterdep).*

El código fuente del programa de prueba (*main*) deberá encontrarse en un fichero llamado *"practica4.cpp"* y **cumplir escrupulosamente con el funcionamiento y formatos que se describieron en la tarea 3 del enunciado de la Práctica 3, o la práctica no será evaluada.** Dado el fichero "entrada.txt", que acompañaba al enunciado de la Práctica 3, el programa para esta Práctica 4 deberá generar un fichero "salida.txt" idéntico al que acompañaba al enunciado de la Práctica 3.

Detalles sobre la implementación con C++:

- **Sobre un *posible* error de compilación en la práctica 4:**
  A la hora de compilar los ficheros de la práctica 4, os podéis encontrar con un error de compilación (por ejemplo) similar a:

```
coleccionInter.hpp:…:….: note:   template argument deduction/substitution failed:
coleccionInter.hpp:93:…: note:   couldn't deduce template parameter  …
            … operacionA(…bla bla…);
```

  Este error nos está diciendo que en la llamada a `operacionA` (en el ejemplo, la llamada a la operación estaría en la línea 93 del fichero coleccionInter.hpp), no se puede deducir qué tipo de dato es el que se quiere usar para los parámetros del template con esa operación.

  El error se soluciona incluyendo explícitamente todos los parámetros para el template en la llamada a la operación genérica `operacionA`, es decir (suponiendo que los parámetros en nuestro template se llaman S y T) escribiendo la llamada a la operación de esta forma:

```
            … operacionA<S,T>(…bla bla…);
```

Observaciones:

- **El código fuente de las prácticas evaluables será compilado y probado en *lab000.unizar.es*, que es donde deberá compilar y funcionar correctamente.** Es recomendable que te asegures de que se podrá compilar y ejecutar correctamente en *lab000* dedicando a ello el tiempo y las pruebas suficientes.
- **El código deberá compilar correctamente con la opción** `-std=c++11` **activada**.
  - Esto significa que, si se trabaja con la línea de comandos, deberá compilarse con:
    `g++ -std=c++11 *.cpp`
- **No se impondrá ninguna ruta concreta para los ficheros de código fuente a compilar, ni para los ficheros que leerá/escribirá el programa**. De esta forma, el fichero de entrada será siempre **"entrada.txt"** (no "entrada2.txt", "datos/entrada.txt" o similares), y el fichero de salida se llamará **"salida.txt"**, no "salida2.txt", "misalida.txt", etc.
- Todos los ficheros con código fuente deberán estar **correctamente documentados**.
- En el **comentario inicial de cada fichero** de código fuente se añadirán los **nombres completos y NIP[1] de las personas autoras de dicho código fuente.**
- La salida debe cumplir escrupulosamente las especificaciones del enunciado. Por ejemplo, cuando se indica que en el fichero de salida se escribirá "`NO INTRODUCIDO: `", está escrito en mayúsculas, con un espacio en blanco tras ':', y lo

[1] El NIP, también llamado NIA; es el número de 6 dígitos que la Universidad de Zaragoza asigna a cada estudiante para su identificación.

mismo para el resto de instrucciones. **Es obligatorio cumplir todos los formatos descritos en este enunciado, o la práctica no será evaluada**.

- **Los TAD deberán implementarse siguiendo las instrucciones dadas en las clases y prácticas de la asignatura, no se permite utilizar Programación Orientada a Objetos, y tampoco el uso o lanzamiento de excepciones.**
- **No se permite usar** las clases o componentes de la Standard Template Library (STL), ni otras bibliotecas similares, y tampoco se permite utilizar *Smart Pointers*. Tampoco se permite utilizar código ajeno, es decir, no implementado **íntegra y personalmente** por las personas integrantes del equipo de prácticas, con la única excepción de la implementación dinámica en C++ del TAD pila suministrada en el material de las clases de la asignatura, y siempre y cuando se utilice reconociendo el origen y autoría original de dicho código.

Entrega de la práctica. Instrucciones:

- **La práctica solo deberá entregarla uno de los miembros del equipo de prácticas**.
- La práctica deberá ser entregada en la tarea correspondiente en el curso Moodle, y antes del plazo límite establecido. **No se aceptarán entregas fuera de plazo, ni entregas por ninguna otra vía que no sea la entrega en la tarea Moodle para esta práctica.**
- Crea un fichero comprimido en formato `Zip` y de nombre `practica4_NIP1_NIP2.zip`, que contenga todos los ficheros de código fuente necesarios para resolver la práctica, y dos ficheros de texto ***entrada.txt*** y ***salida.txt***, con los formatos descritos en el enunciado, pero que sean significativamente diferentes a los proporcionados como ejemplo, y con los que habréis probado la implementación realizada en vuestra práctica. Los ficheros no deben estar distribuidos en subdirectorios, y no debe entregarse ningún fichero de código objeto, ni ejecutables, etc.
- El nombre utilizado para el fichero comprimido: `practica4_NIP1_NIP2.zip`, debe contener los NIP (también llamados NIA) de ambos integrantes del equipo, siendo `NIP1<NIP2`. Si el trabajo se ha hecho de forma individual el nombre del fichero deberá ser: `practica4_NIP1.zip` , siendo `NIP1` el NIP de la persona que ha realizado la práctica.
- A la hora de evaluar la práctica se utilizará tanto el fichero de prueba *entrada.txt* y *salida.txt* que se entreguen, como ficheros de prueba entregados por otros compañeros, o ficheros propios de los profesores.